
STARRFILMS.COM

Quantum Computing Engineers Set New Standard in Silicon Chip Performance

Engineers at Australia's University of New South Wales, Sydney (UNSW Sydney) have coaxed quantum computing processors to hold data for up to two milliseconds, a more than 100-fold increase over previous benchmarks. This achievement extends the researchers' successful manipulation of millions of quantum bits (qubits) with a single antenna last year. … UNSW Sydney's Henry Yang called the SMART protocol "a potential path for full-scale quantum computers."

https://newsroom.unsw.edu.au/news/science-tech/longest-time-quantum-computing-engineers-set-new-standard-silicon-chip-performance

UC Berkeley
Teaching Professor
Dan Garcia

CS61C

Great Ideas in Computer Architecture

(a.k.a. Machine Structures)

UC Berkeley
Lecturer
Justin Yokota

Pipelining

New-School Machine Structures

Software

Parallel Requests
Assigned to computer
e.g., Search "Cats"

Parallel Threads
Assigned to core e.g., Lookup, Ads

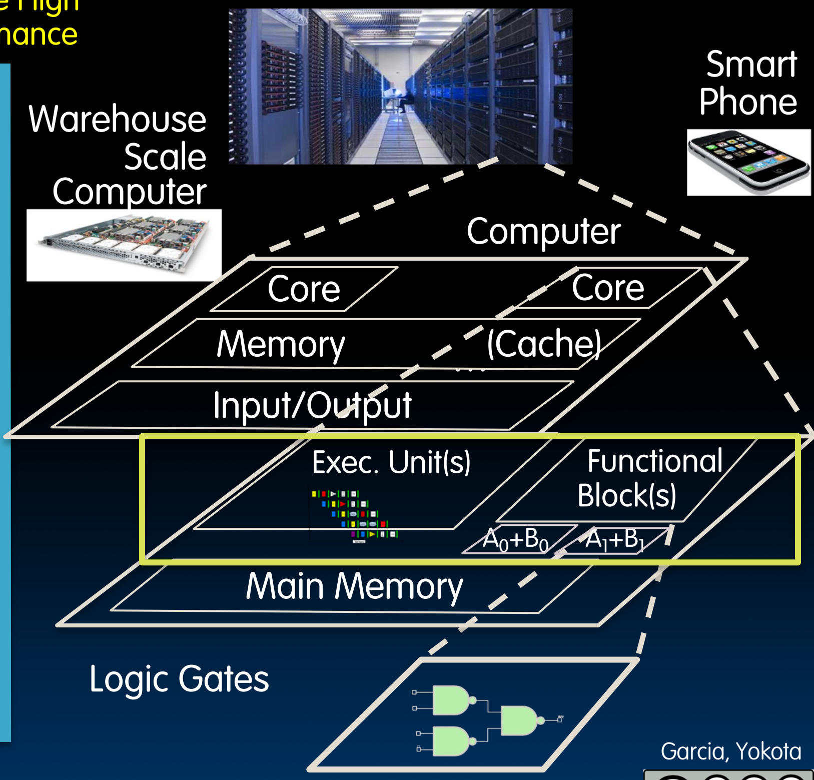
Parallel Instructions
>1 instruction @ one time
e.g., 5 pipelined instructions

Parallel Data
>1 data item @ one time
e.g., Add of 4 pairs of words

Hardware descriptions
All gates work in parallel at same time

Harness Parallelism & Achieve High Performance

Hardware

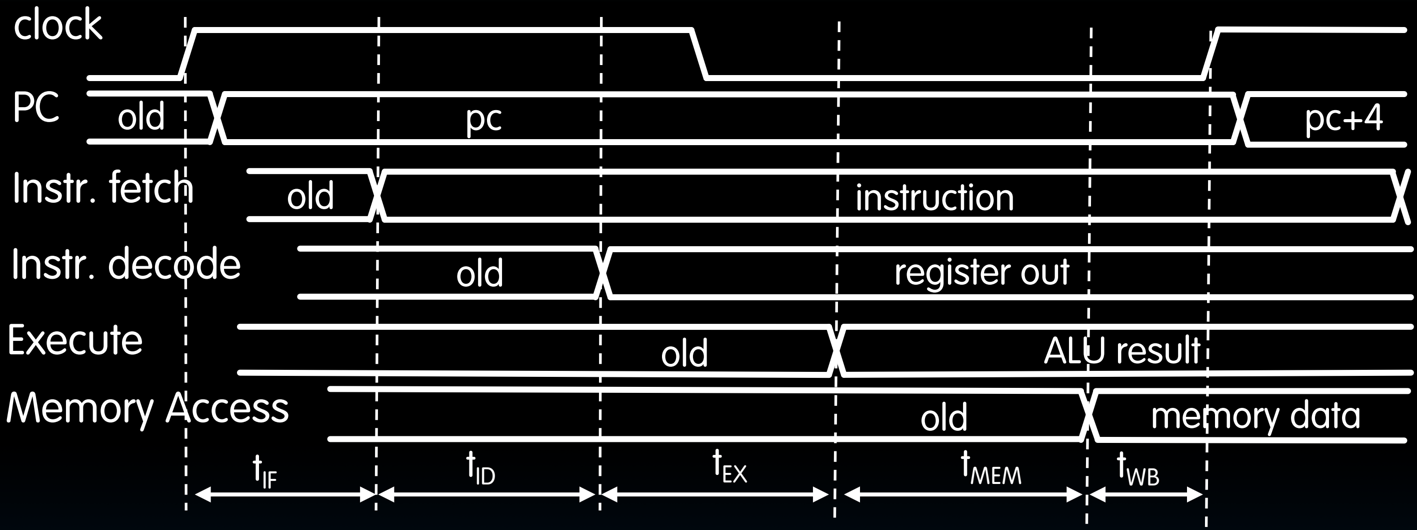


6 Great Ideas in Computer Architecture

1. Abstraction (Layers of Representation/Interpretation)
2. Moore's Law
3. Principle of Locality/Memory Hierarchy
4. Parallelism
5. Performance Measurement & Improvement
6. Dependability via Redundancy

Garcia, Yokota

Instruction Timing



IF	ID	EX	MEM	WB	Total
I-MEM	Reg Read	ALU	D-MEM	Reg W	
200 ps	100 ps	200 ps	200 ps	100 ps	800 ps

Garcia, Yokota

Instruction Timing

Instr	IF = 200ps	ID = 100ps	ALU = 200ps	MEM=200ps	WB = 100ps	Total
`add`	X	X	X		X	600ps
`beq`	X	X	X			500ps
`jal`	X	X	X			500ps
`lw`	X	X	X	X	X	800ps
`sw`	X	X	X	X		700ps

- Maximum clock frequency
  - f_{max} = 1/800ps = 1.25 GHz

Garcia, Yokota

Performance Measures

- "Our" Single-cycle RISC-V CPU executes instrs at 1.25 GHz
  - 1 instruction every 800 ps
- Can we improve its performance?
  - What do we mean with this statement?
  - Not so obvious:
    - Quicker response time, so one job finishes faster?
    - More jobs per unit time (e.g. web server returning pages, spoken words recognized)?
    - Longer battery life?

Transportation Analogy

	Sports Car	Bus
Passenger Capacity	2	50
Travel Speed	200 mph	50 mph
Gas Mileage	5 mpg	2 mpg

50 Mile trip (assume they return instantaneously)

	Sports Car	Bus
Travel Time	15 min	60 min
Time for 100 passengers	750 min (50 2-person trips)	120 min (2 50-person trips)
Gallons per passenger	5 gallons	0.5 gallons

Garcia, Yokota

Computer Analogy

Transportation	Computer
Trip Time	Program execution time: e.g. time to update display
Time for 100 passengers	Throughput: e.g. number of server requests handled per hour
Gallons per passenger	Energy per task*: e.g. how many movies you can watch per battery charge or energy bill for datacenter

* Note: Power is not a good measure, since low-power CPU might run for a long time to complete one task consuming more energy than faster computer running at higher power for a shorter time

Processor Performance Iron Law

“Iron Law” of Processor Performance

$$\frac{\text{Time}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} * \frac{\text{Cycles}}{\text{Instruction}} * \frac{\text{Time}}{\text{Cycle}}$$

CPI = Cycles Per Instruction

Instructions per Program

Determined by

- Task
- Algorithm, e.g. $O(N^2)$ vs $O(N)$
- Programming language
- Compiler
- Instruction Set Architecture (ISA)

Garcia, Yokota

(Average) Clock Cycles per Instruction (CPI)

Determined by

- ISA
- Processor implementation (or microarchitecture)
- E.g. for "our" single-cycle RISC-V design, CPI = 1
- Complex instructions (e.g. `strcpy`), CPI >> 1
- Superscalar processors, CPI < 1 (next lectures)

Time per Cycle (1/Frequency)

Determined by

- Processor microarchitecture (determines critical path through logic gates)
- Technology (e.g. 5nm versus 28nm)
- Power budget (lower voltages reduce transistor speed)

Garcia, Yokota

Speed Tradeoff Example

- For some task (e.g. image compression) …

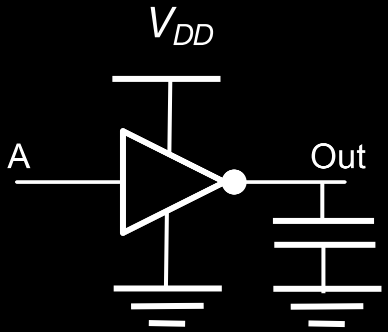
	Processor A	Processor B
# Instructions	1 Million	1.5 Million
Average CPI	2.5	1
Clock rate f	2.5 GHz	2 GHz
Execution time	1 ms	0.75 ms

Processor B is faster for this task, despite executing more instructions and having a slower clock rate!

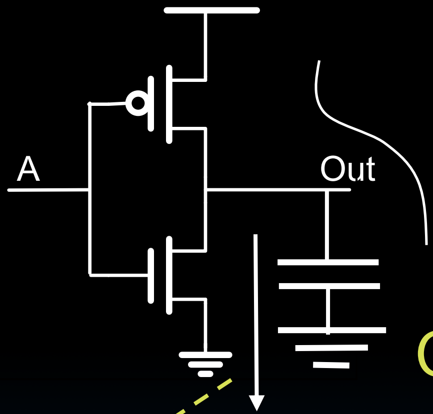
Energy Efficiency

Where Does Energy Go in CMOS?

Symbol (INV)



Schematic



Leakage

Charging capacitors (CV^2)

(30%)

(70%)

Energy per Task

$$\frac{\text{Energy}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} * \frac{\text{Energy}}{\text{Instruction}}$$

$$\frac{\text{Energy}}{\text{Program}} \propto \frac{\text{Instructions}}{\text{Program}} * C \; V^2$$

"Capacitance" depends on technology, processor features e.g. # of cores

Supply voltage, e.g. 1V

Want to reduce capacitance and voltage to reduce energy/task

Garcia, Yokota

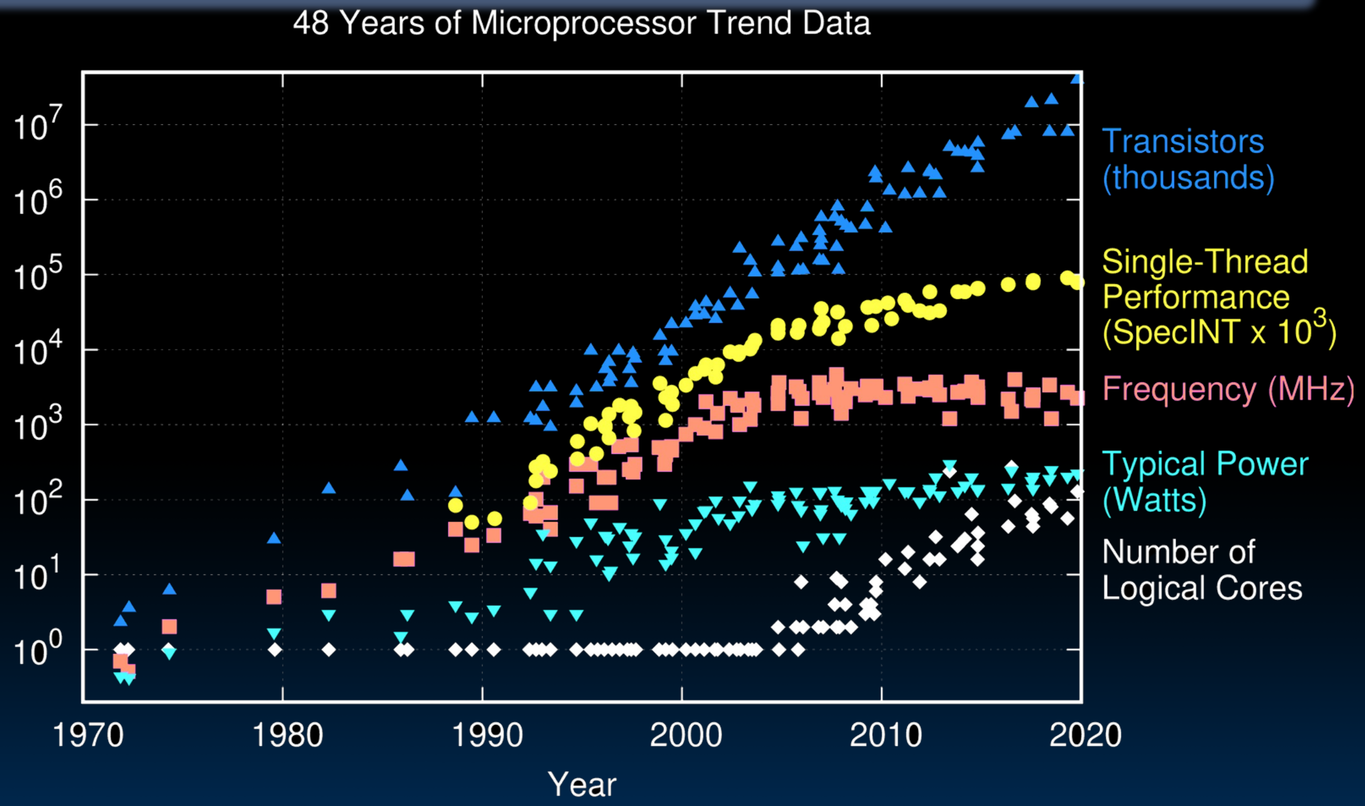
Energy Tradeoff Example

- "Next-generation" processor
  - C (Moore's Law): -15 %
  - Supply voltage, Vsup: -15 %
  - Energy consumption: 0 - (1-0.85[3]) = -39 %

- Significantly improved energy efficiency thanks to
  - Moore's Law AND
  - Reduced supply voltage

Garcia, Yokota

Performance/Power Trends



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2019 by K. Rupp

End of Dennard Scaling

- In 1974, Robert Dennard observed that power density remained constant for a given area of silicon, while the dimension of the transistor shrank
- In recent years, industry has not been able to reduce supply voltage much, as reducing it further would mean increasing "leakage power" where transistor switches don't fully turn off (more like dimmer switch than on-off switch)
- Also, size of transistors and hence capacitance, not shrinking as much as before between transistor generations
  - Need to go to 3D
- Power becomes a growing concern – the "power wall"

Energy "Iron Law"

Performance = Power * Energy Efficiency

(Tasks/Second) (Joules/Second) (Tasks/Joule)

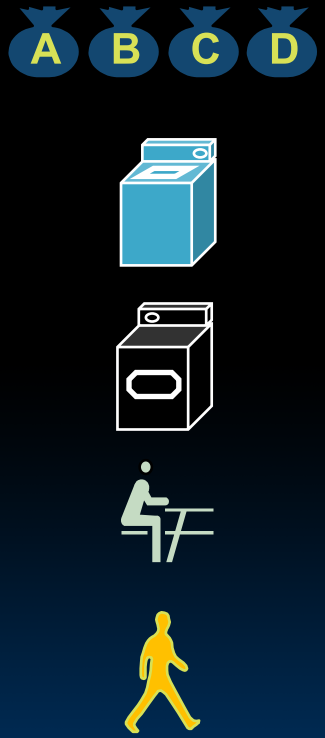
- Energy efficiency (e.g., instructions/Joule) is key metric in all computing devices
- For power-constrained systems (e.g., 20MW datacenter), need better energy efficiency to get more performance at same power
- For energy-constrained systems (e.g., 1W phone), need better energy efficiency to prolong battery life

Garcia, Yokota

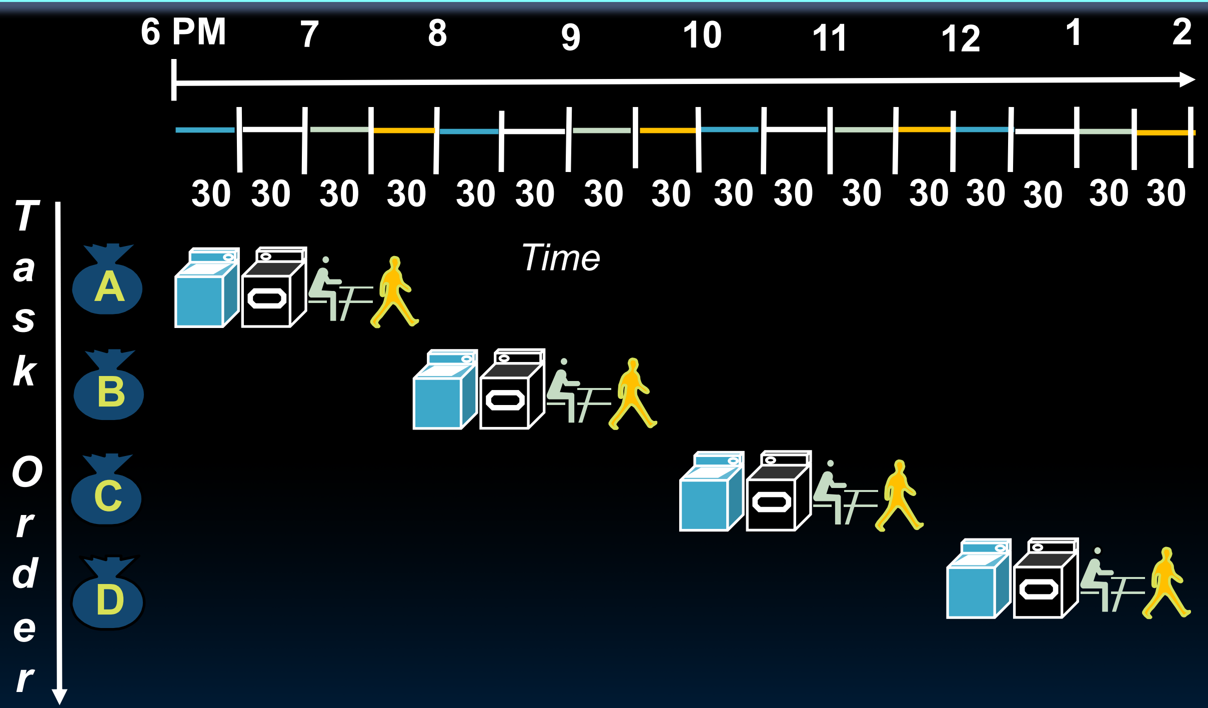
Introduction to Pipelining

Gotta Do Laundry

- Avi, Bora, Caroline, Dan
  each have one load of clothes to wash, dry, fold, and put away
  - Washer takes 30 minutes
  - Dryer takes 30 minutes
  - "Folder" takes 30 minutes
  - "Stasher" takes 30 minutes to put clothes into drawers

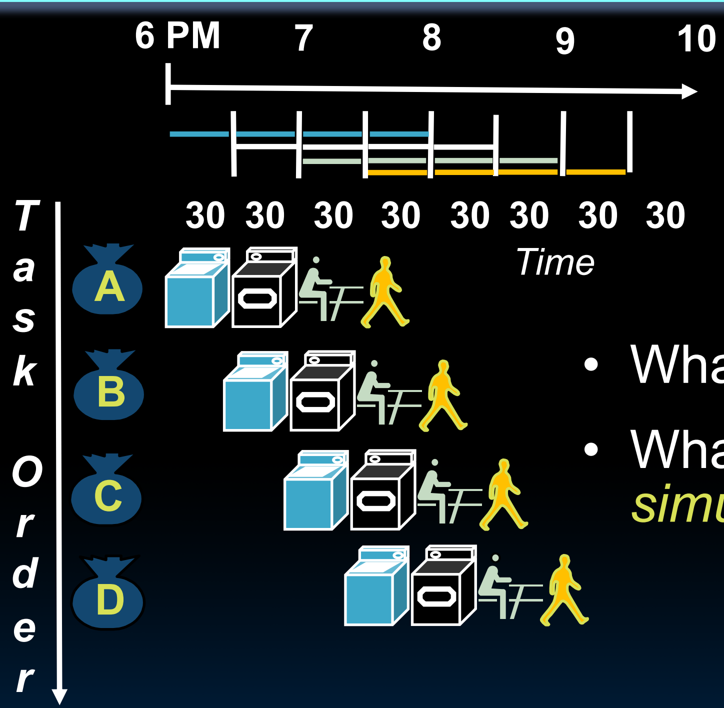


Sequential Laundry



Sequential laundry takes 8 hours for 4 loads!

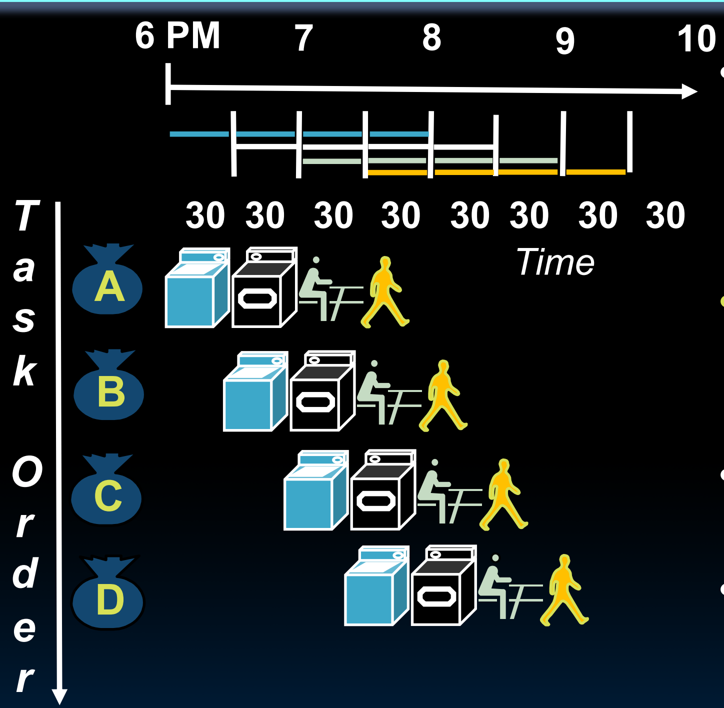
Pipelined Laundry



- What happens *sequentially*?
- What happens *simultaneously*?

Pipelined laundry takes 3.5 hours for 4 loads!

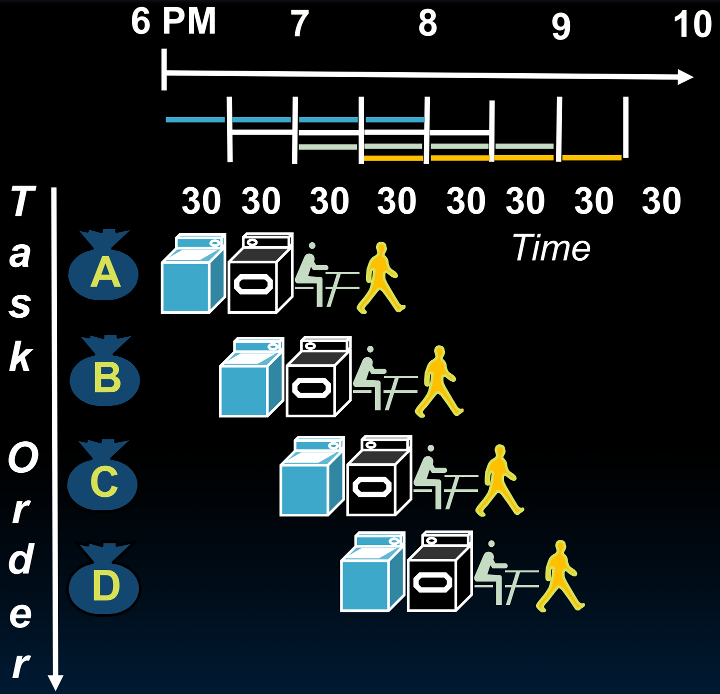
Sequential Laundry



- Pipelining doesn't help latency of single task, it helps throughput of entire workload
- Multiple tasks operating simultaneously using different resources
- Potential speedup = Number of pipe stages
- Time to "fill" pipeline and time to "drain" it reduces speedup: 2.3X v. 4X in this example

Garcia, Yokota

Sequential Laundry



Suppose:
- new Washer takes 20 minutes
- new Stasher takes 20 minutes.

How much faster is pipeline?

Pipeline rate limited by slowest pipeline stage

Unbalanced lengths of pipe stages reduce speedup

When poll is active, respond at **pollev.com/ddg**

Text **DDG** to **22333** once to join

L21 Thanks to pipelining, I have reduced the time it took me to wash my shirt.

Longer pipelines are always a win (since less work per stage & a faster clock).

A processor's CPI remains relatively flat (say ± 25%) across different benchmarks

FFF
FFT
FTF
FTT
TFF
TFT
TTF
TTT

And in Conclusion, ...

- Instruction timing
  - Set by instruction complexity, architecture, technology
  - Pipelining increases clock frequency, "instructions per second"
    - But does not reduce time to complete instruction
- Performance measures
  - Different measures depending on objective
    - Response time
    - Jobs / second
    - Energy per task