CS 250
VLSI System Design

Lecture 3 – Timing

2014-9-4

Professor Jonathan Bachrach

slides by John Lazzaro

TA: Colin Schmidt

www-inst.eecs.berkeley.edu/~cs250/

UC Regents Fall 2013/1014 © UCB

... everything doesn't happen at once.

- **Timing, the 10,000 ft view.** Locally synchronous, globally asynchronous.
- **On the same page.** Minimal set of timing concepts you need for project.

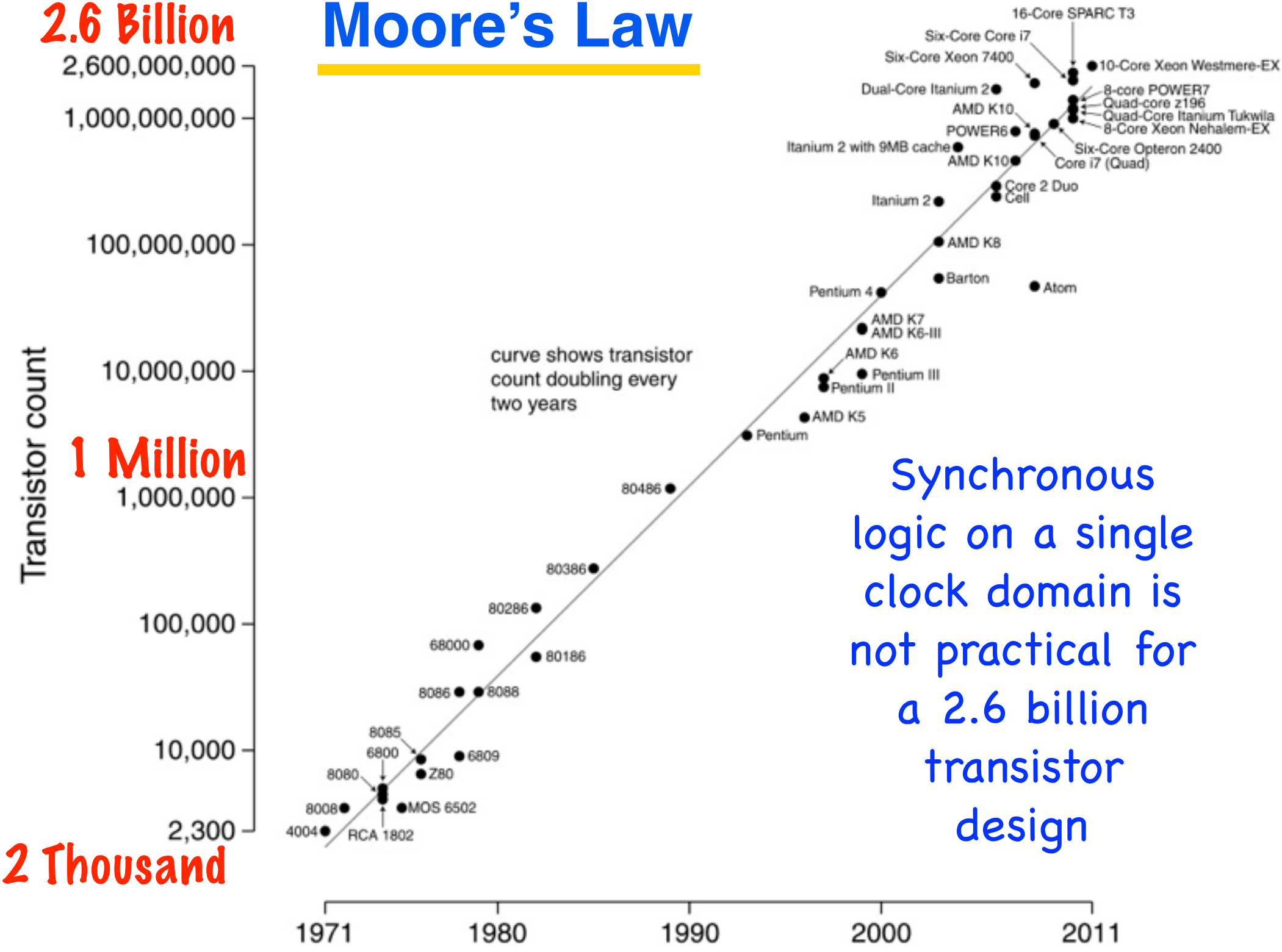
Break

- **RTL Examples.** Better timing through micro-architecture.
- **Electrical details.** Just so you know ...

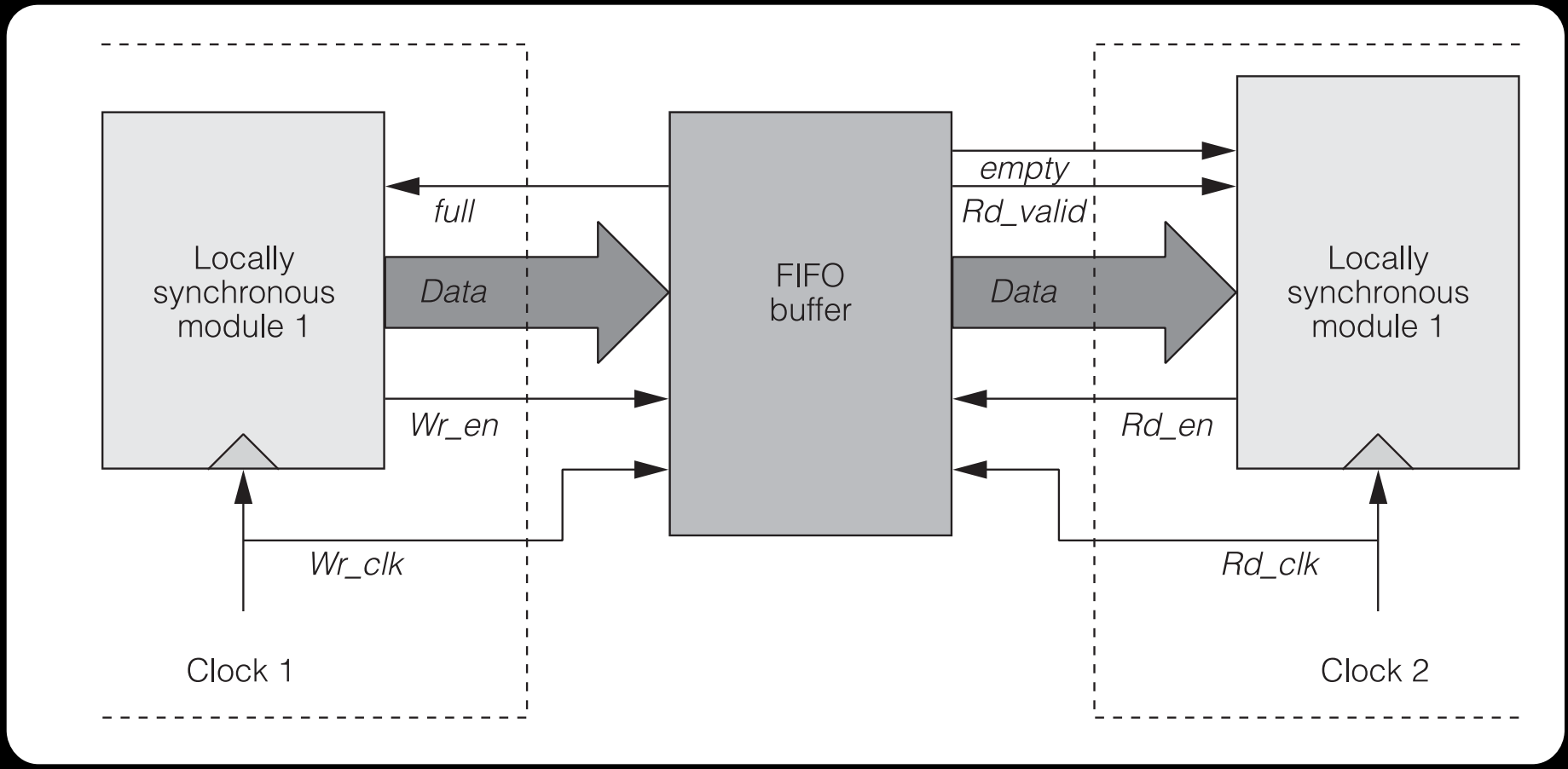
UC Regents Fall 2013 © UCB

View from 10,000 Ft.
Google I/O, 2012

Moore's Law

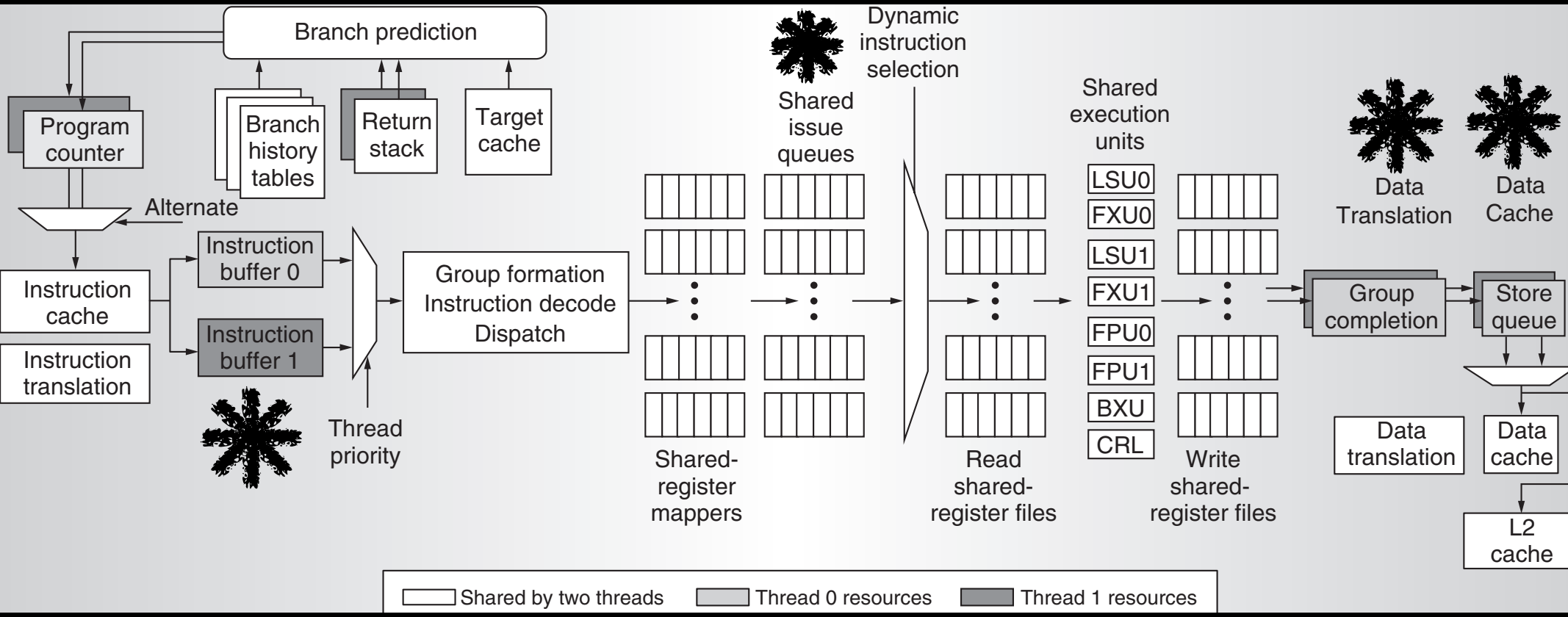

2.6 Billion
1 Million
2 Thousand
Transistor count
2,600,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
2,300
1971
1980
1990
2000
2011
curve shows transistor count doubling every two years
16-Core SPARC T3
Six-Core Core i7
Six-Core Xeon 7400
10-Core Xeon Westmere-EX
Dual-Core Itanium 2
8-core POWER7
Quad-core z196
AMD K10
Quad-Core Itanium Tukwila
POWER6
8-Core Xeon Nehalem-EX
Itanium 2 with 9MB cache
Six-Core Opteron 2400
AMD K10
Core i7 (Quad)
Core 2 Duo
Cell
Itanium 2
AMD K8
Barton
Atom
Pentium 4
AMD K7
AMD K6-III
AMD K6
Pentium III
Pentium II
AMD K5
Pentium
80486
80386
80286
68000
80186
8086
8088
8085
6800
6809
Z80
8080
8008
MOS 6502
4004
RCA 1802
Synchronous logic on a single clock domain is not practical for a 2.6 billion transistor design

GALS: Globally Asynchronous, Locally Synchronous



Synchronous modules typically 50K-1M gates, so that the synchronous logic approach works well without requiring heroics. Examples ...

IBM Power 5 CPU - Dynamically Scheduled

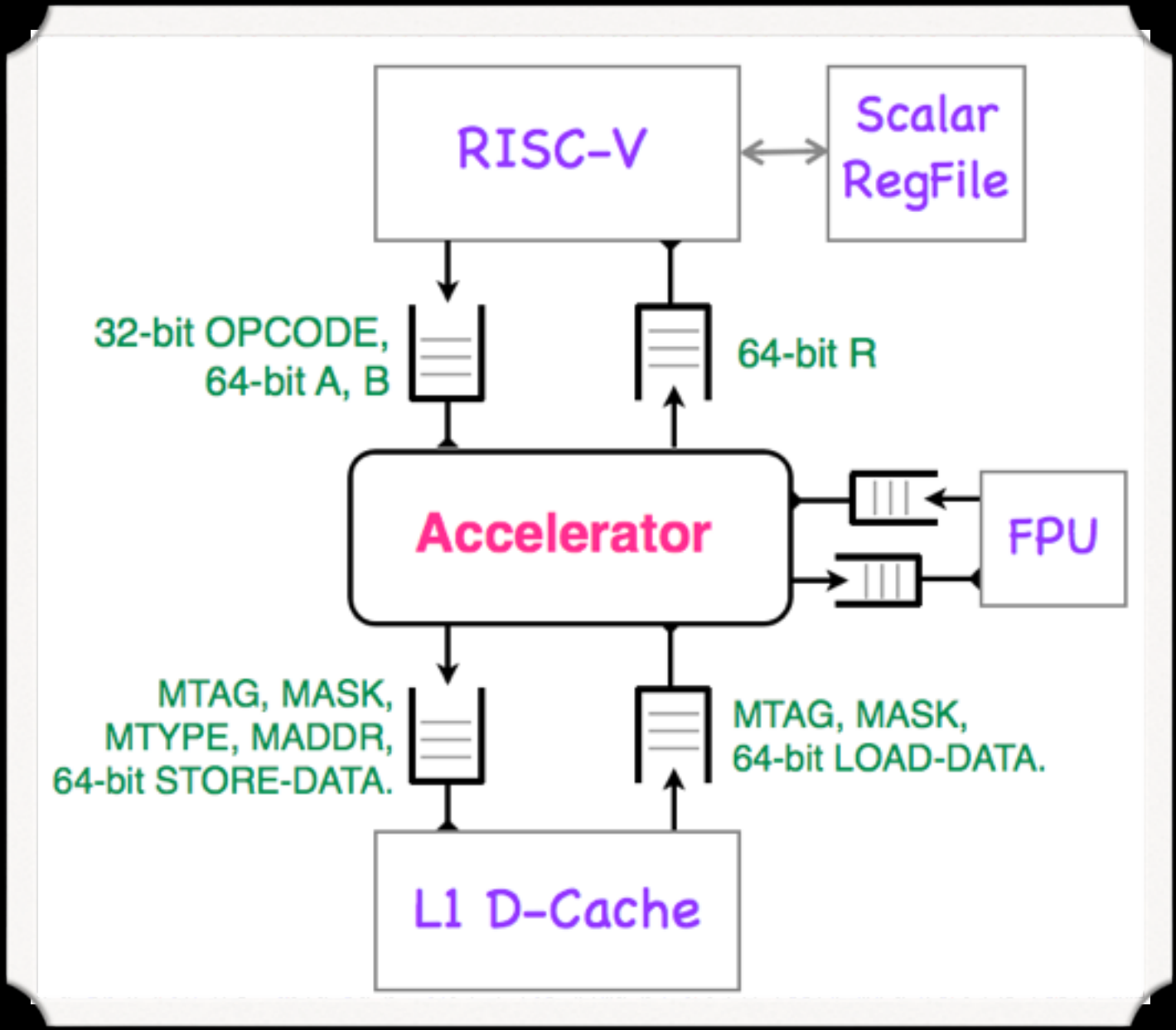


Stars denote FIFOs that create separate synchronous domains. An example of how architecture and circuits work together.

Rocket uses GALS for accelerator interface

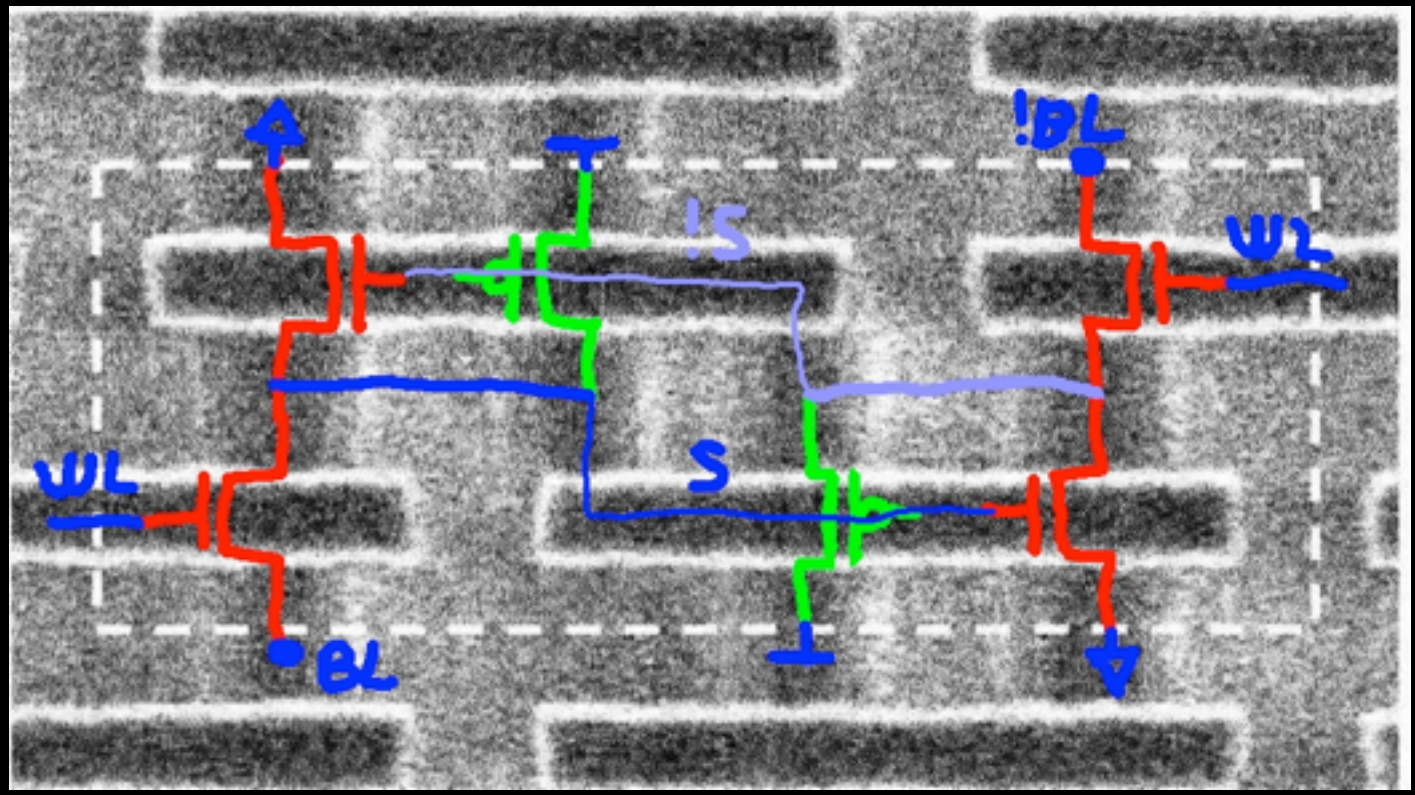
Your project interfaces with the RISC-V pipeline and the memory system using FIFOs.

Your timing closure is independent of the CPU logic domain.



Today: Timing insights for your project

What we're not doing. If this class was EE 241 and your project was an SRAM:



You could see through down to the layout.

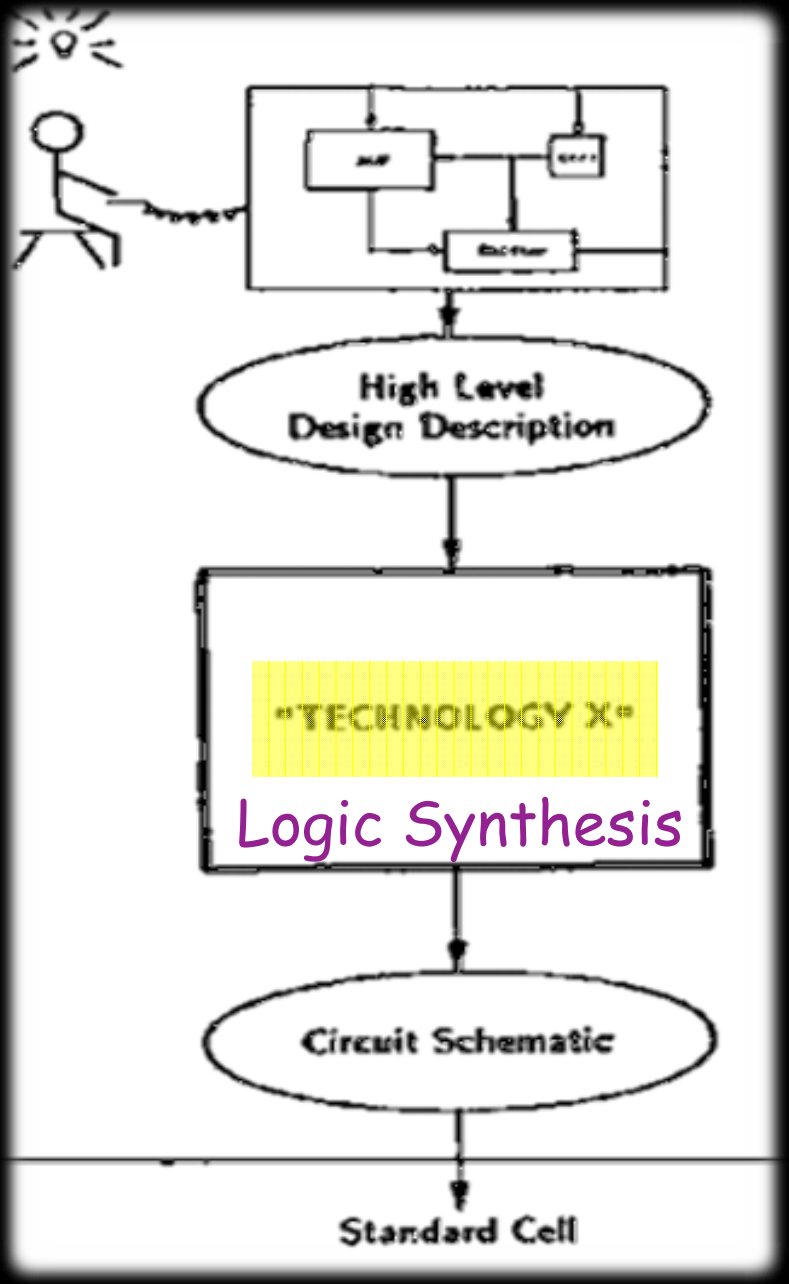
Timing? Use SPICE on this hand-drawn schematic.

Technology X: The CS 250 timing challenge.

- What we are doing --->
- If your accelerator is too slow ... two options:

Top-down: Rework high-level micro-architecture. Let Technology X keep its job. **Today.**

Bottom-up: Take control away from logic synthesis. Use HDL as textual schematic. Also, use command-line tool flags.



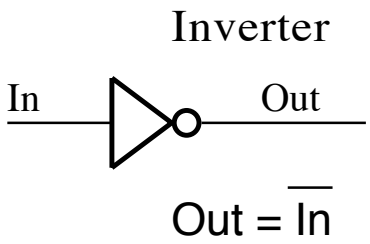
Sometimes necessary. Colin is the expert, ask in discussion

A Logic Circuit Primer

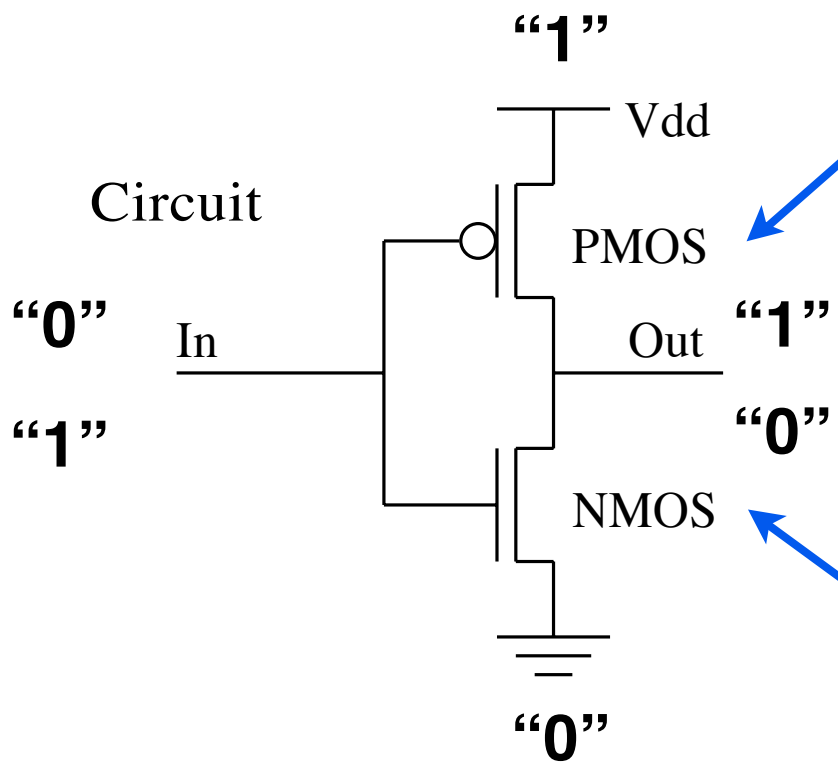
"Models should be as simple as possible,
but no simpler ..." Albert Einstein.

UC Regents Fall 2013/1014 © UCB

Inverters: A simple transistor model



In	Out
0	1
1	0



pFET.
A switch.
"On" if
gate is
grounded.

Correctly predicts logic output for simple static CMOS circuits.

nFET.
A switch.
"On" if
gate is
at Vdd.

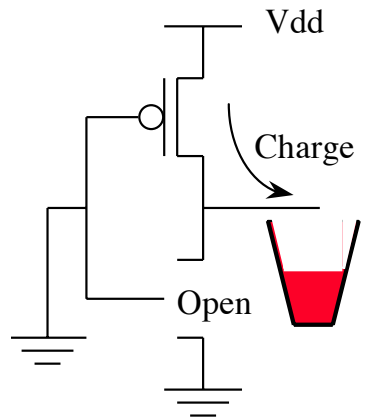
Extensions to model subtler circuit families, or to predict timing, have not worked well ...

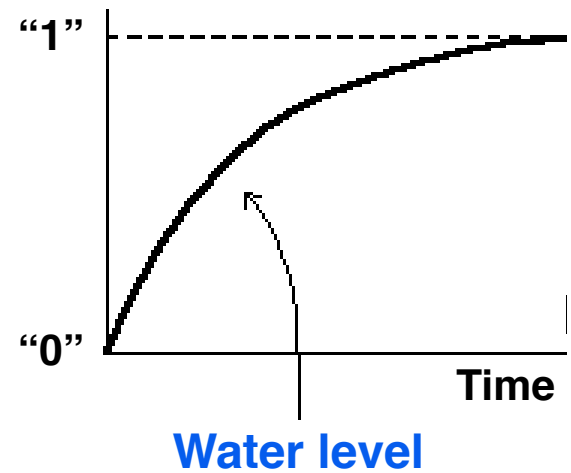
UC Regents Fall 2013 © UCB

Transistors as water valves. (Cartoon physics)

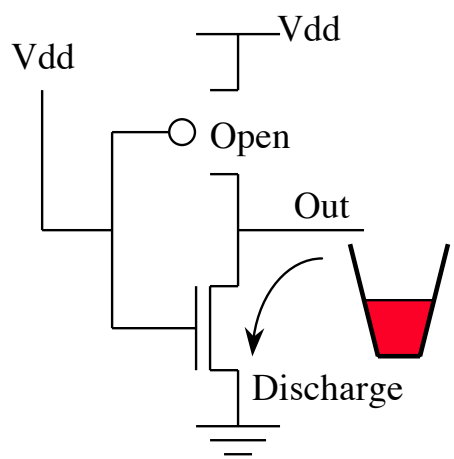
If electrons are water molecules, transistor strengths (W/L) are pipe diameters, and capacitors are buckets ...

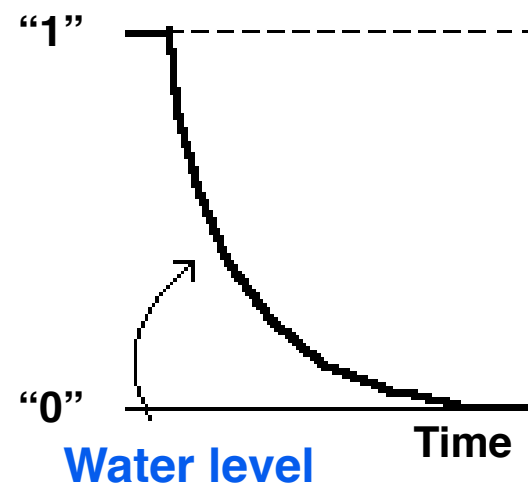
A "on" p-FET fills up the capacitor with charge.





A "on" n-FET empties the bucket.



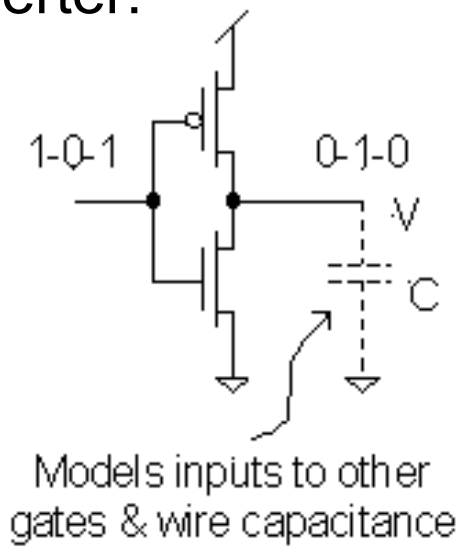


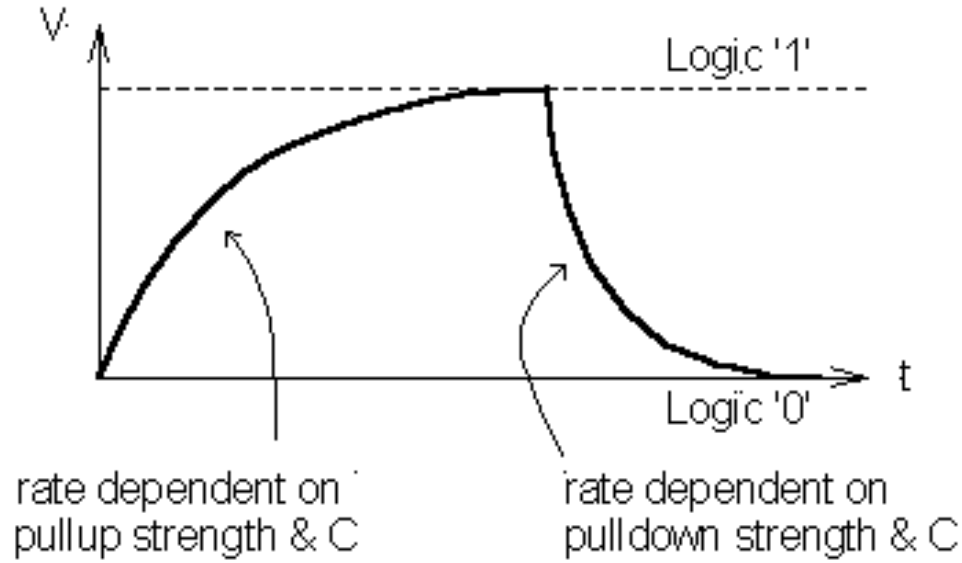
This model is often good enough ...

UC Regents Fall 2013/2014 © UCB

What is the bucket? A gate's "fan-out".

Inverter:





"Fan-out": The number of gate inputs driven by a gate's output.

Driving other gates slows a gate down.

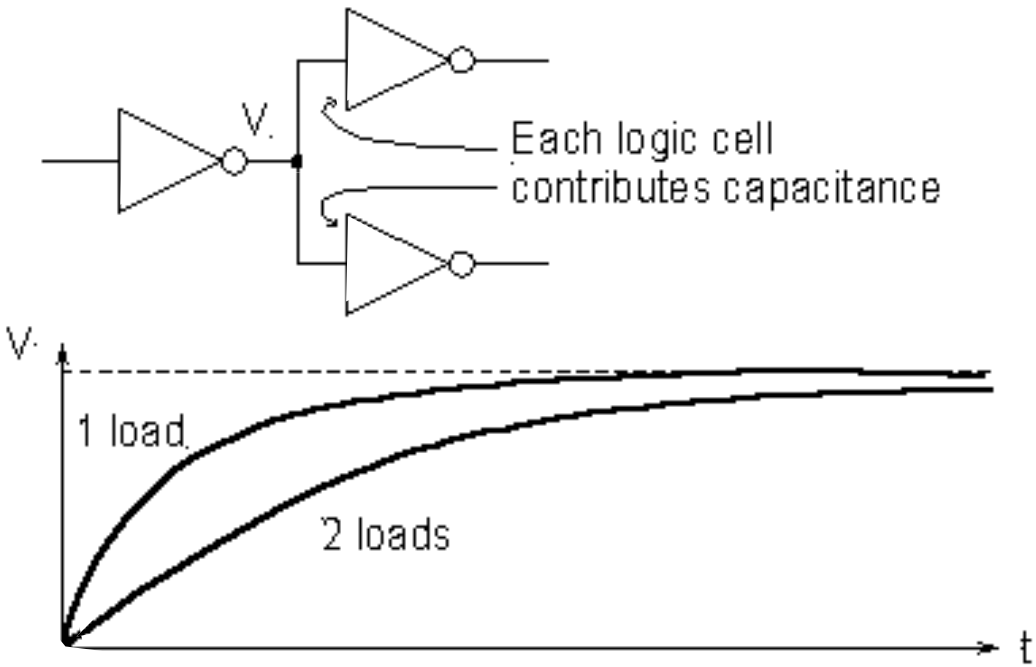
Driving wires slows a gate down.

Driving it's own parasitics slows a gate down.

UC Regents Fall 2013 © UCB

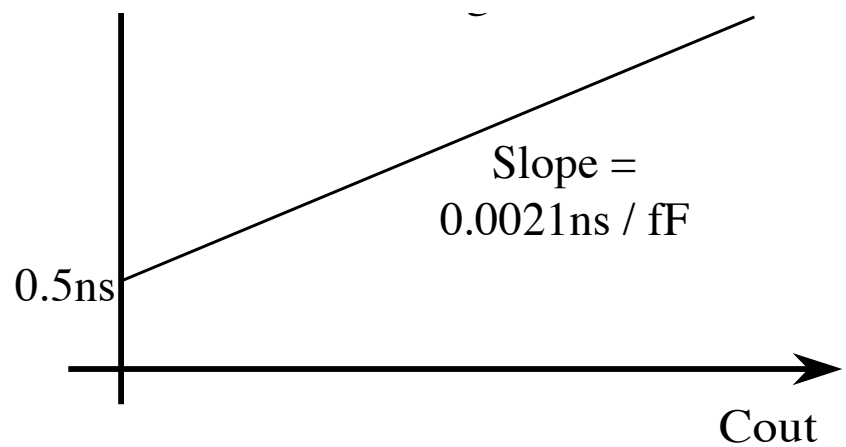
Fanout

A closer look at fan-out ...

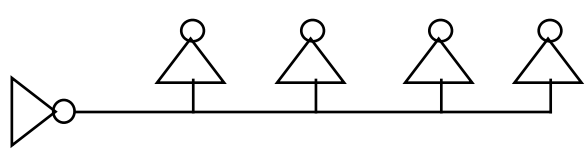


Driving more gates adds delay.

Linear model works for reasonable fan-out



FO4: Fanout of four delay.

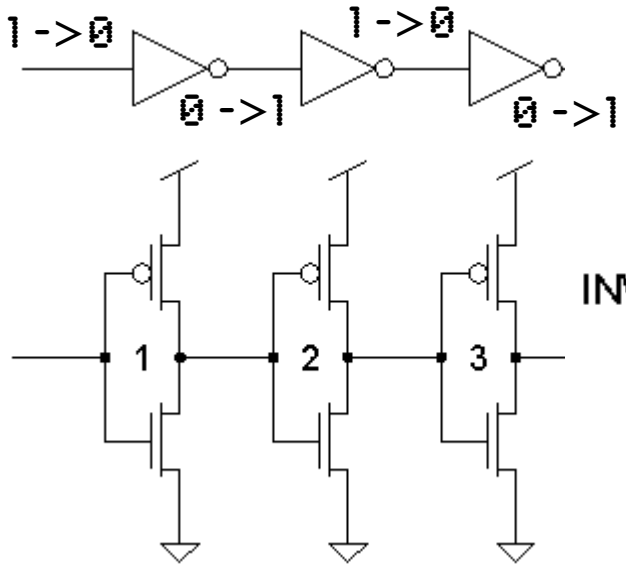


Delay time of an inverter driving 4 inverters.

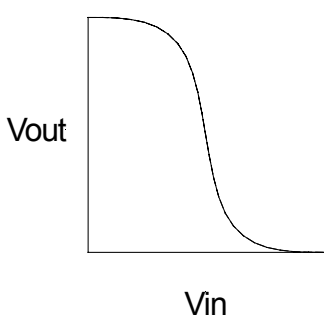
UC Regents Fall 2013 © UCB

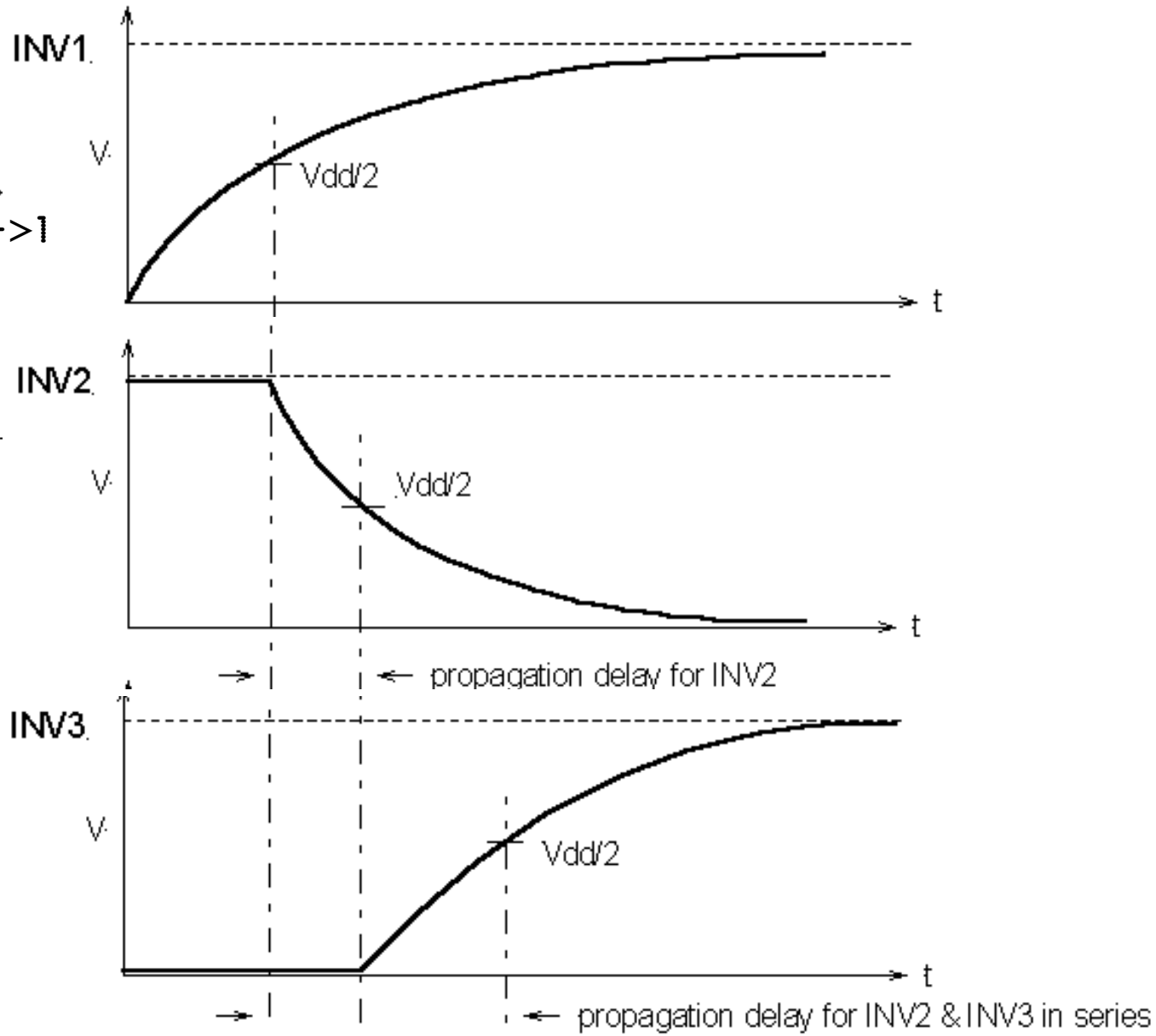
Propagation delay graphs ...

- Cascaded gates:



INVERTER TRANSFER FUNCTION

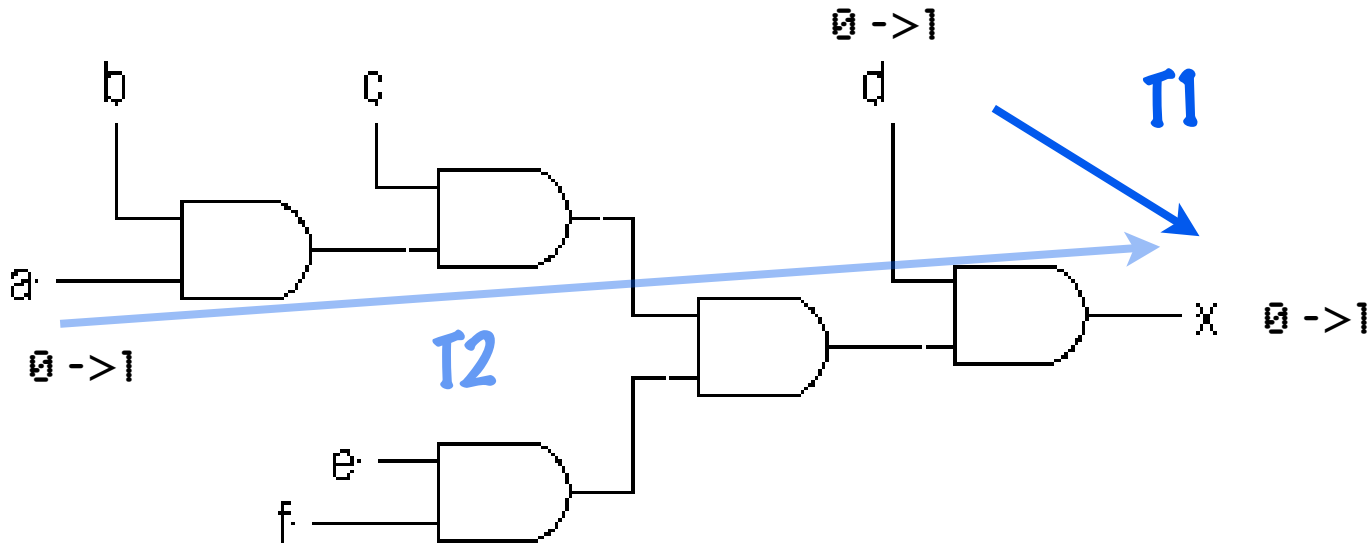




UC Regents Fall 2013/2014 © UCB

Worst-case delay through combinational logic

T2 might be the worst-case delay path (critical path)



$$x = g(a, b, c, d, e, f)$$

If d going 0-to-1 switches x 0-to-1, delay is T1.

If a going 0-to-1 switches x 0-to-1, delay is T2.

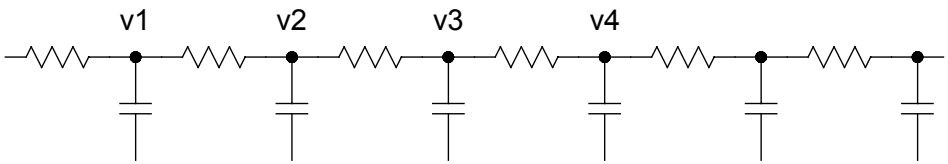
It would be surprising if T1 > T2.

UC Regents Fall 2013/2014 © UCB

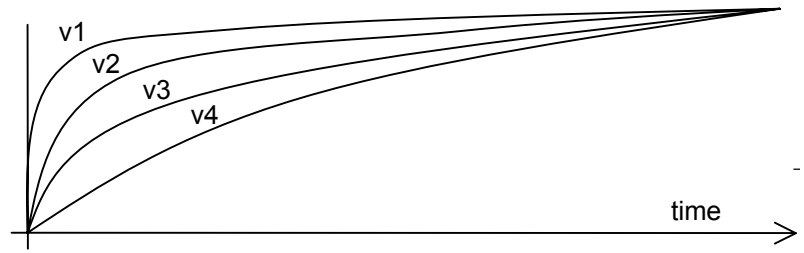
Why “might”? Wires have delay too ...

Looks benign, but ...

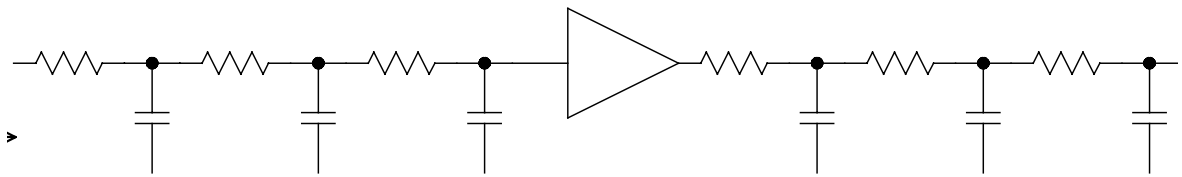
– Wires posses distributed resistance and capacitance



– Time constant associated with distributed RC is proportional to the *square* of the length



– signals are typically “rebuffered” to reduce delay:

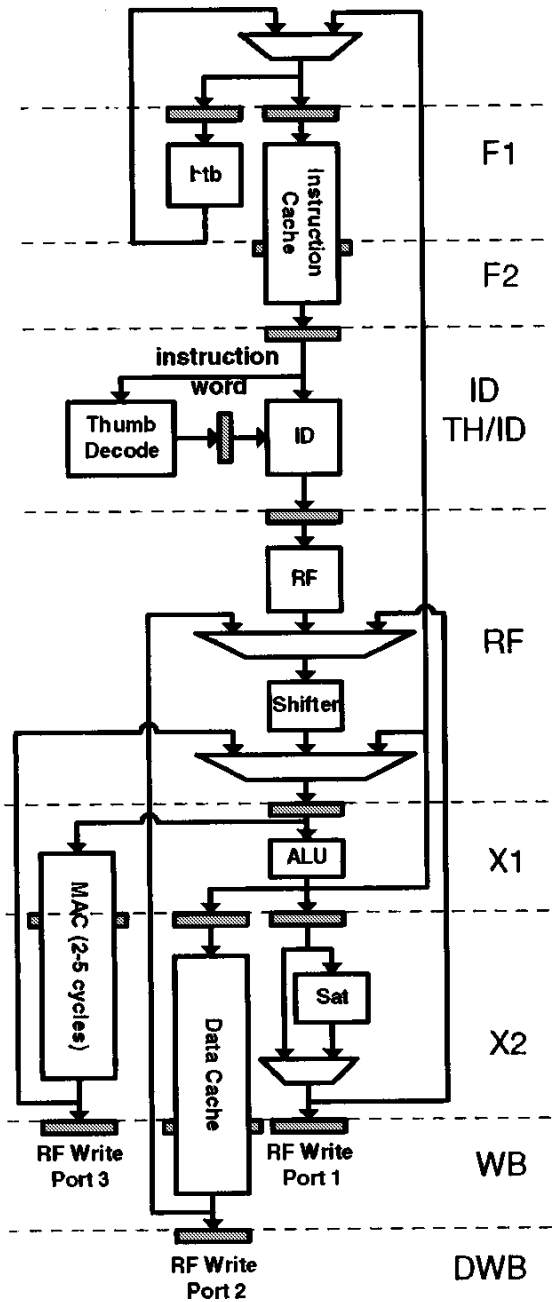


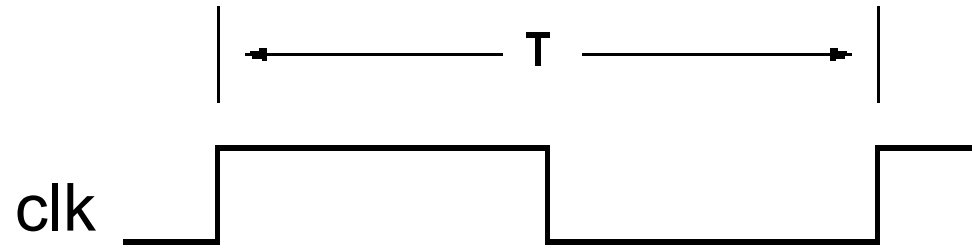
UC Regents Fall 2013/2014 © UCB

Clocked Logic Circuits

UC Regents Fall 2013/1014 © UCB

From Delay Models to Timing Analysis





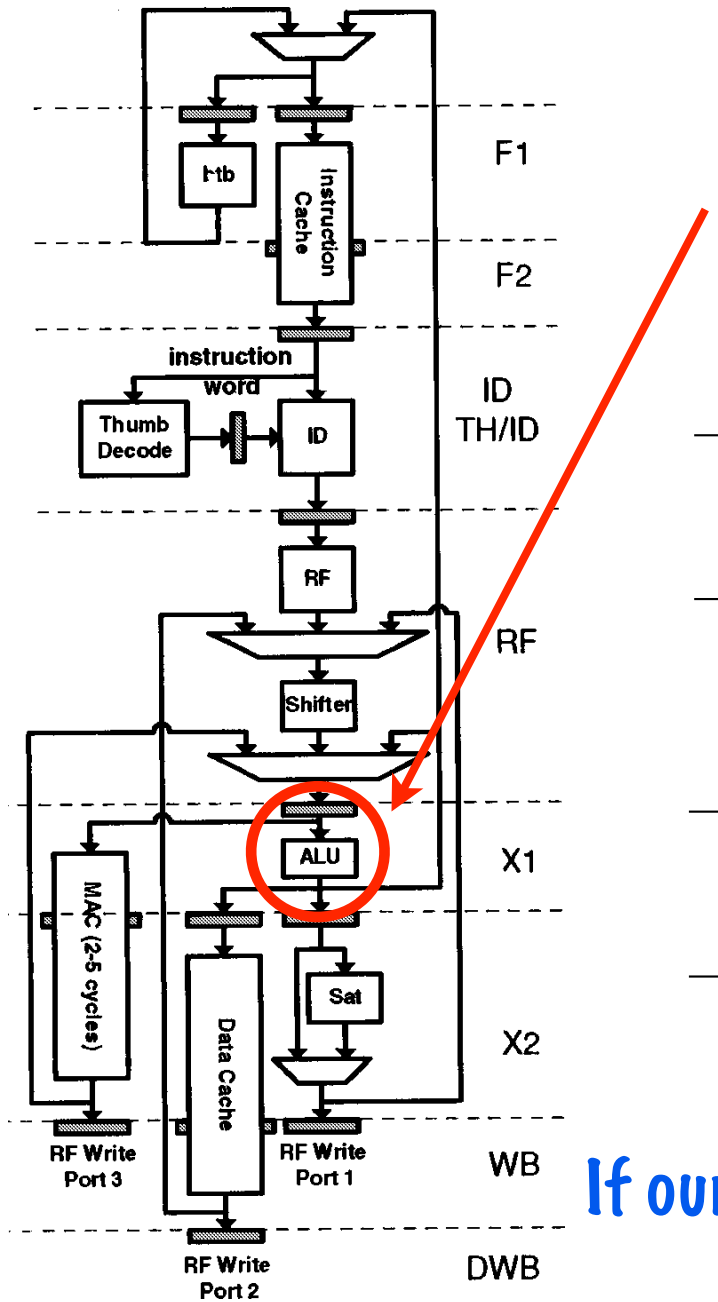
Timing Analysis

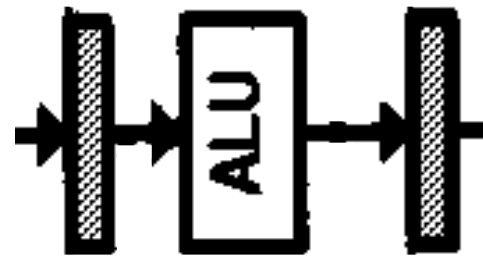
What is the smallest T that produces correct operation?

f	T
1 MHz	1 µs
10 MHz	100 ns
100 MHz	10 ns
1 GHz	1 ns

UC Regents Fall 2013/2014 © UCB

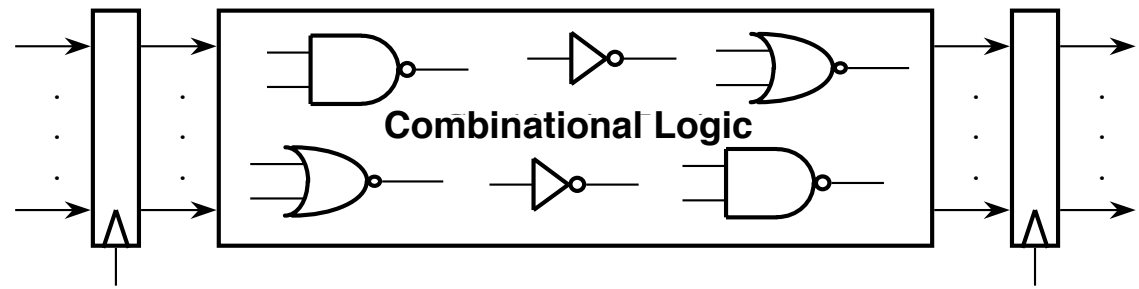
Timing Analysis and Logic Delay

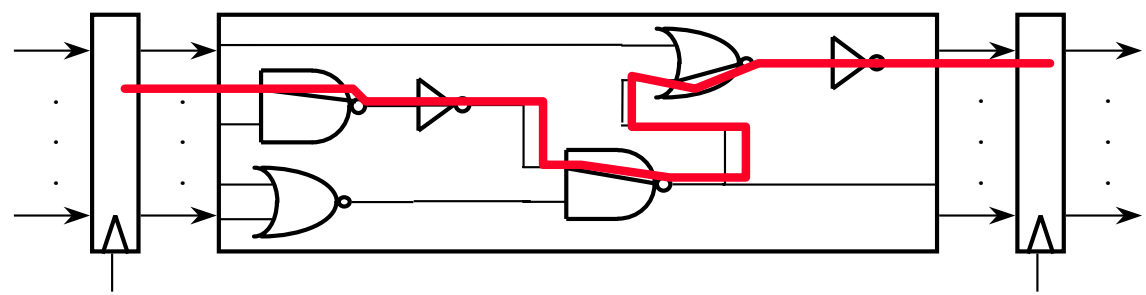




Register:

An Array of Flip-Flops

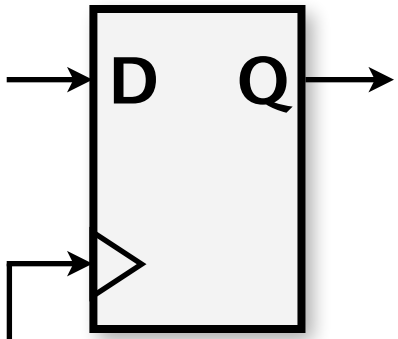




If our clock period T > worst-case delay through CL, does this ensure correct operation?

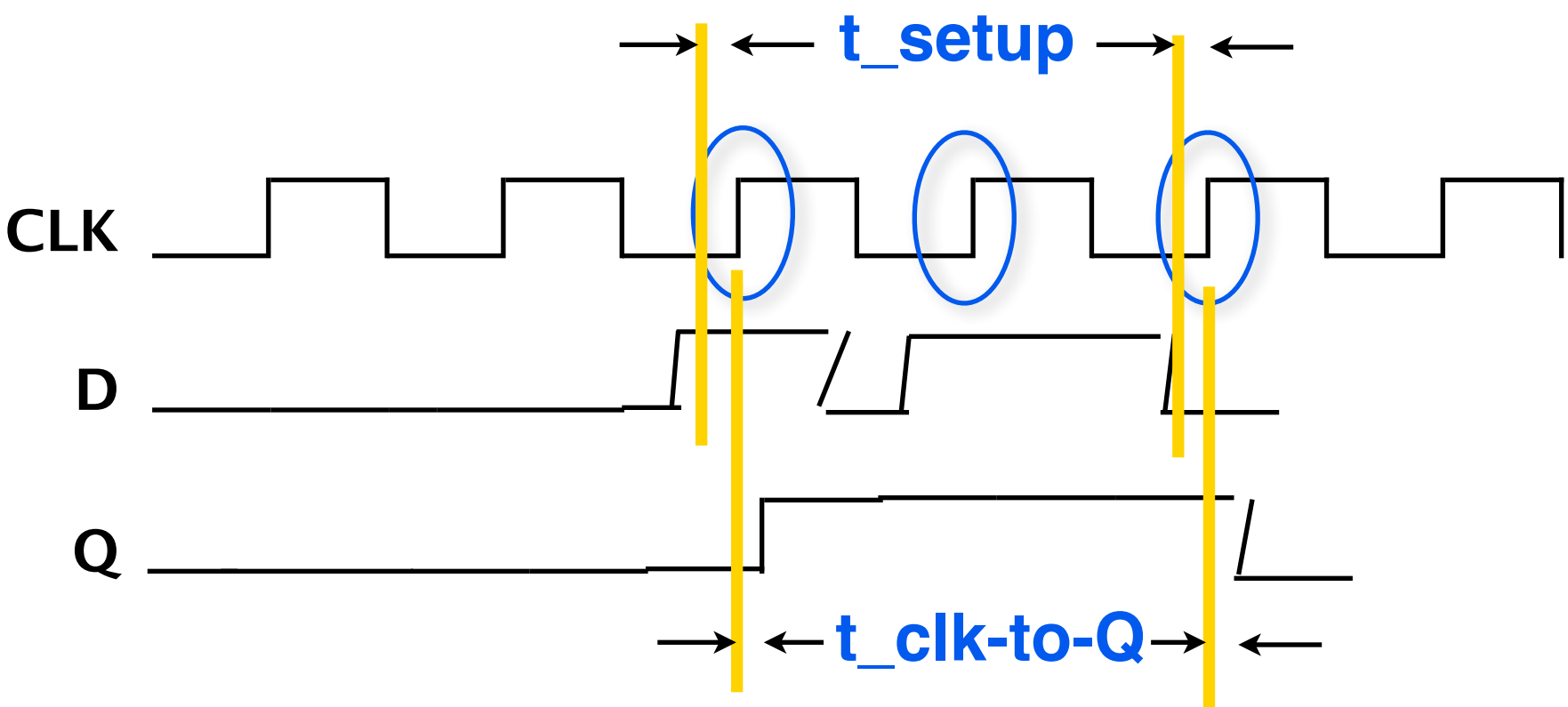
UC Regents Fall 2013/2014 © UCB

Flip Flops have internal delays ...



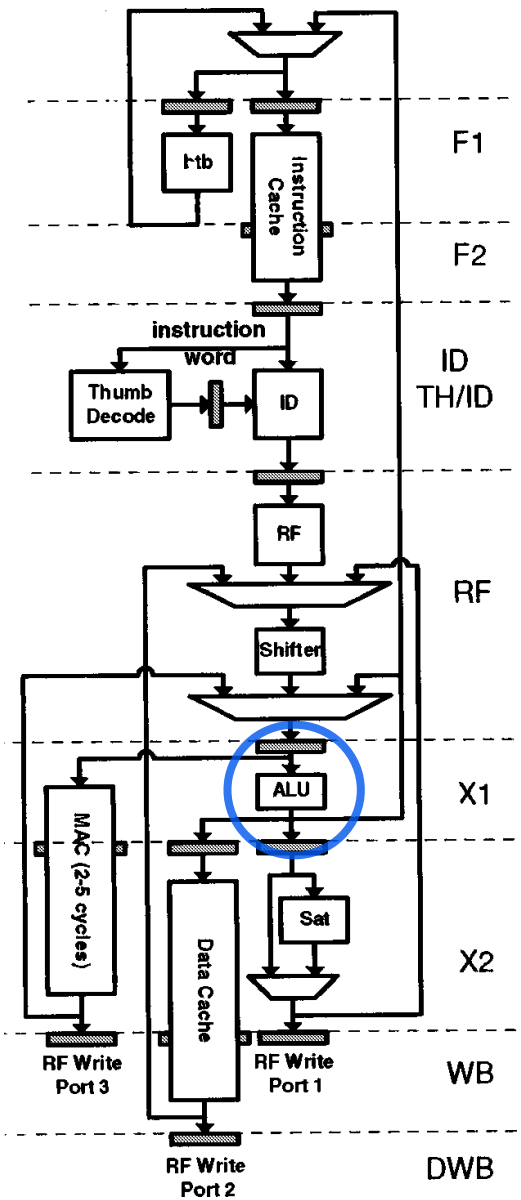
Value of D is sampled on positive clock edge.

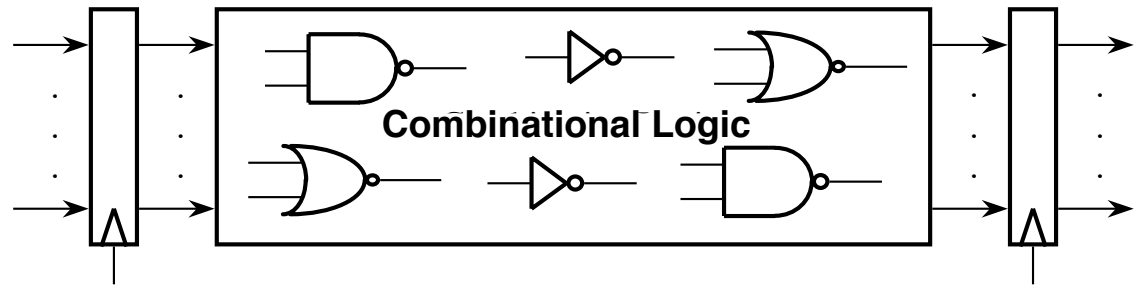
Q outputs sampled value for rest of cycle.

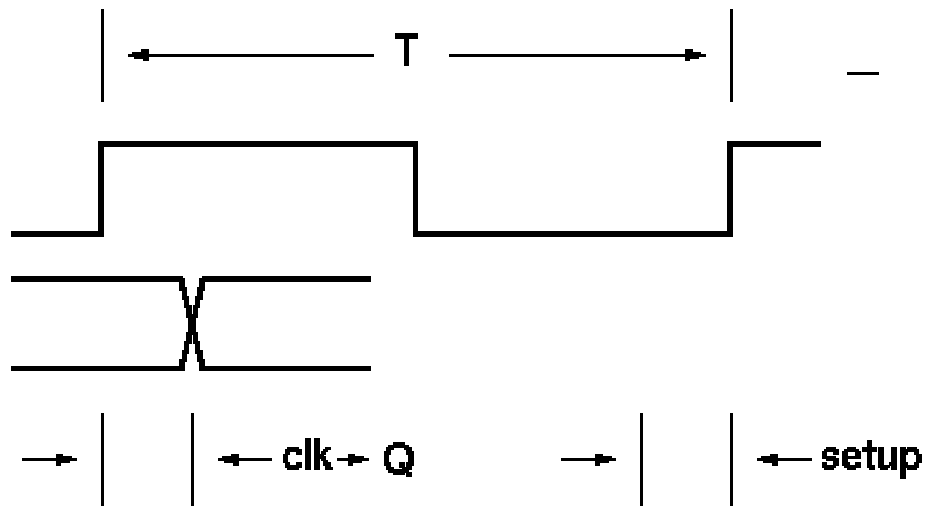


UC Regents Fall 2013/2014 © UCB

Flip-Flop delays eat into "time budget"





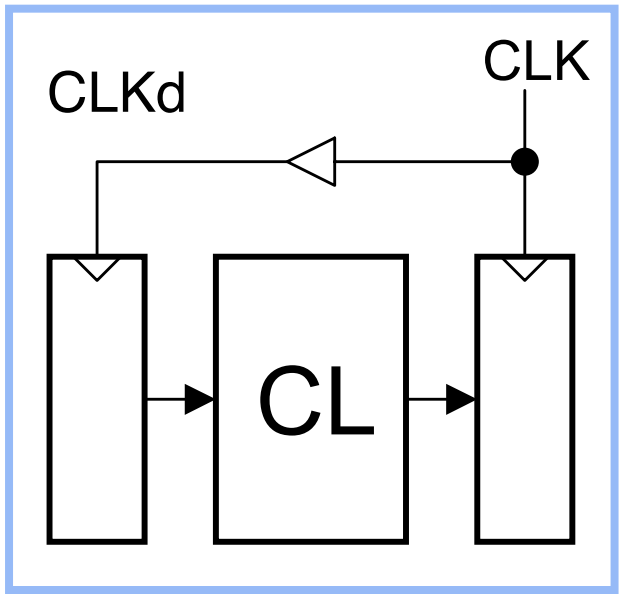


ALU "time budget"

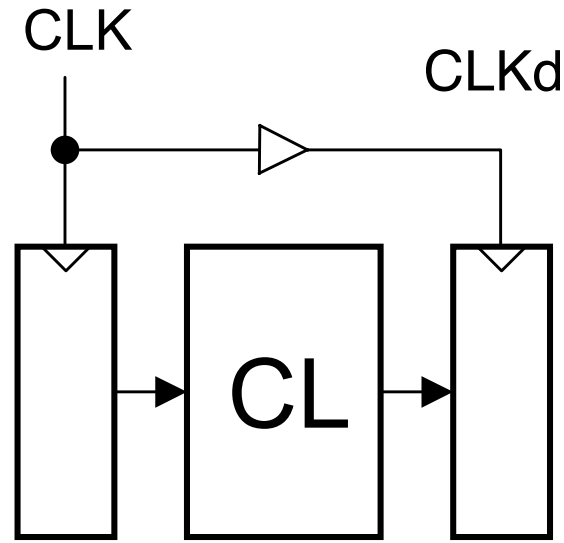
$$T \geq \tau_{clk \rightarrow Q} + \tau_{CL} + \tau_{setup}$$

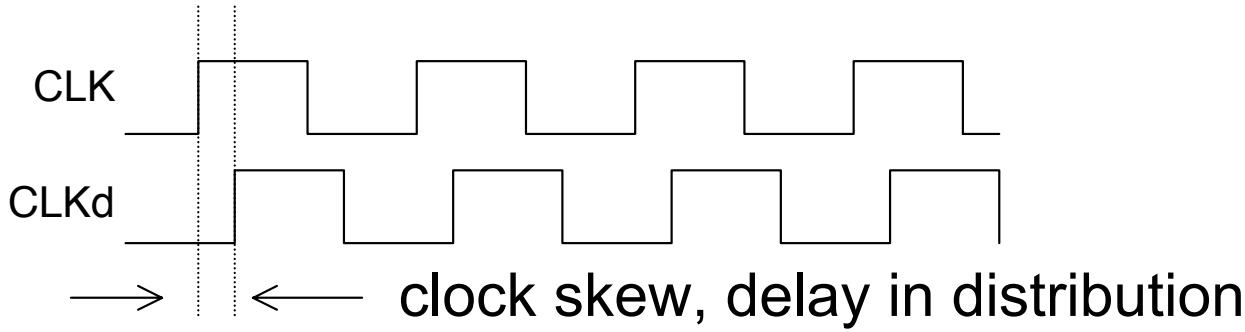
UC Regents Fall 2013/2014 © UCB

Clock skew also eats into "time budget"



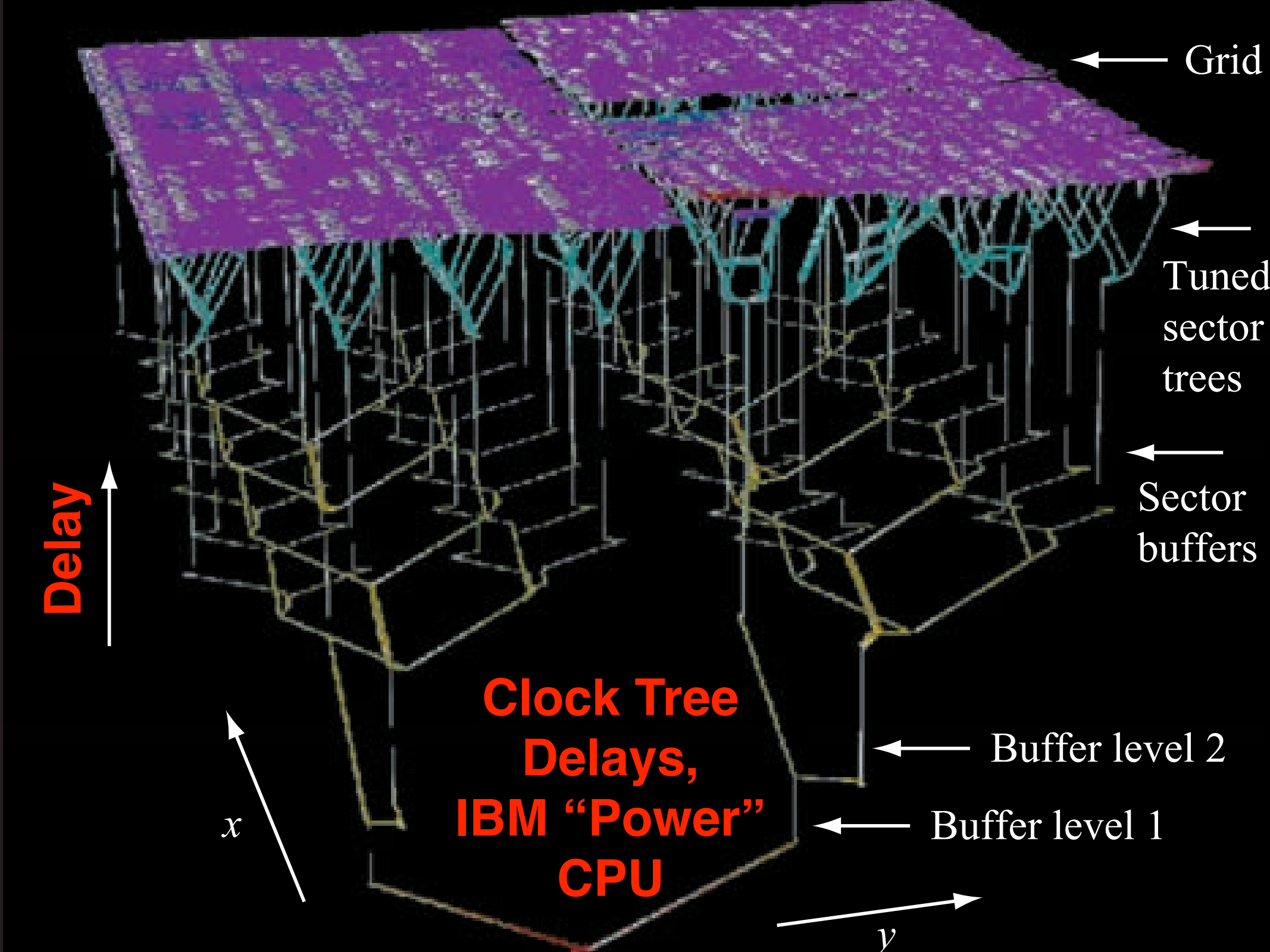
As T →0, which circuit fails first?

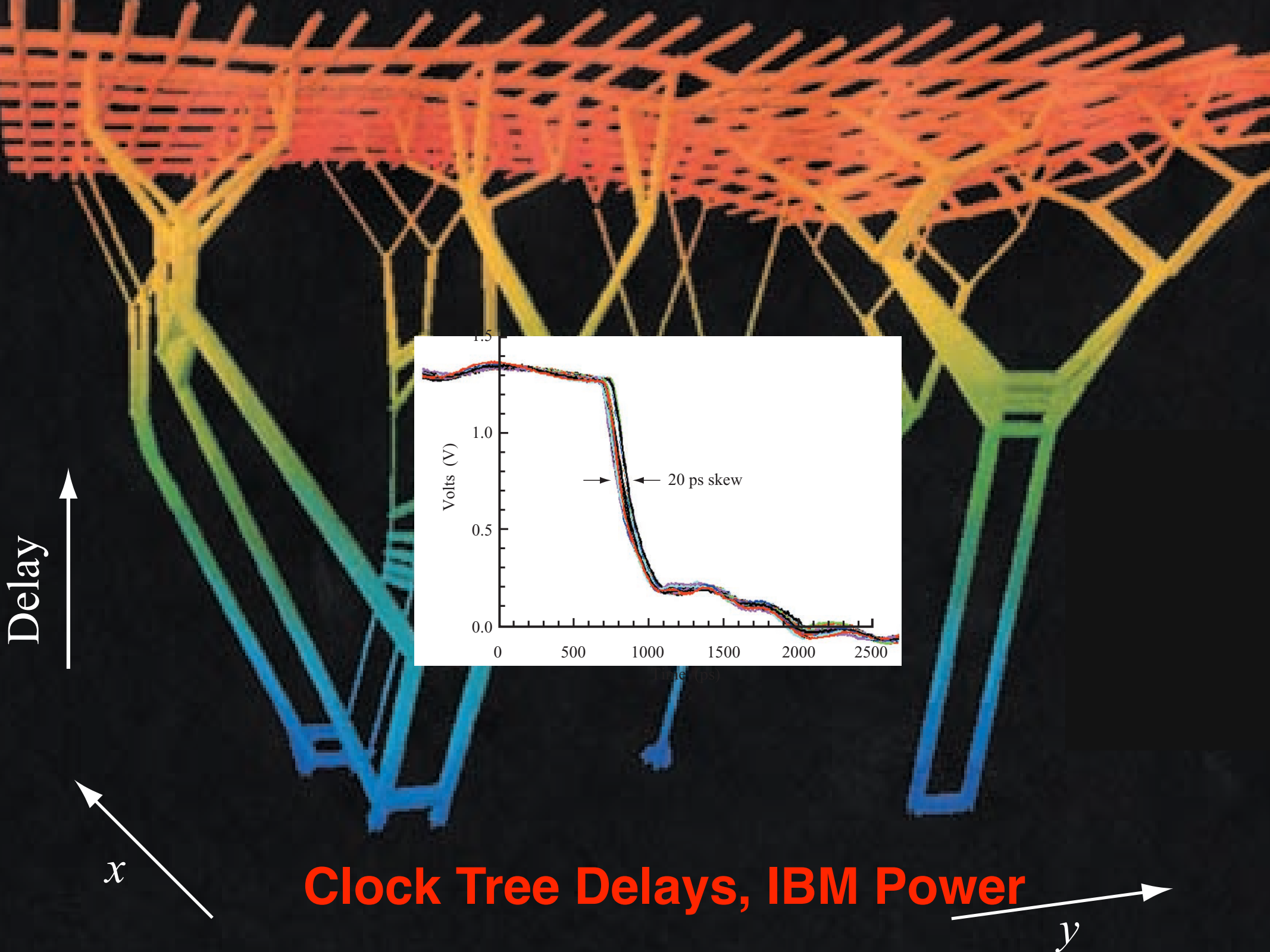




$$T \geq T_{CL} + T_{setup} + T_{clk \rightarrow Q} + \text{worst case skew.}$$

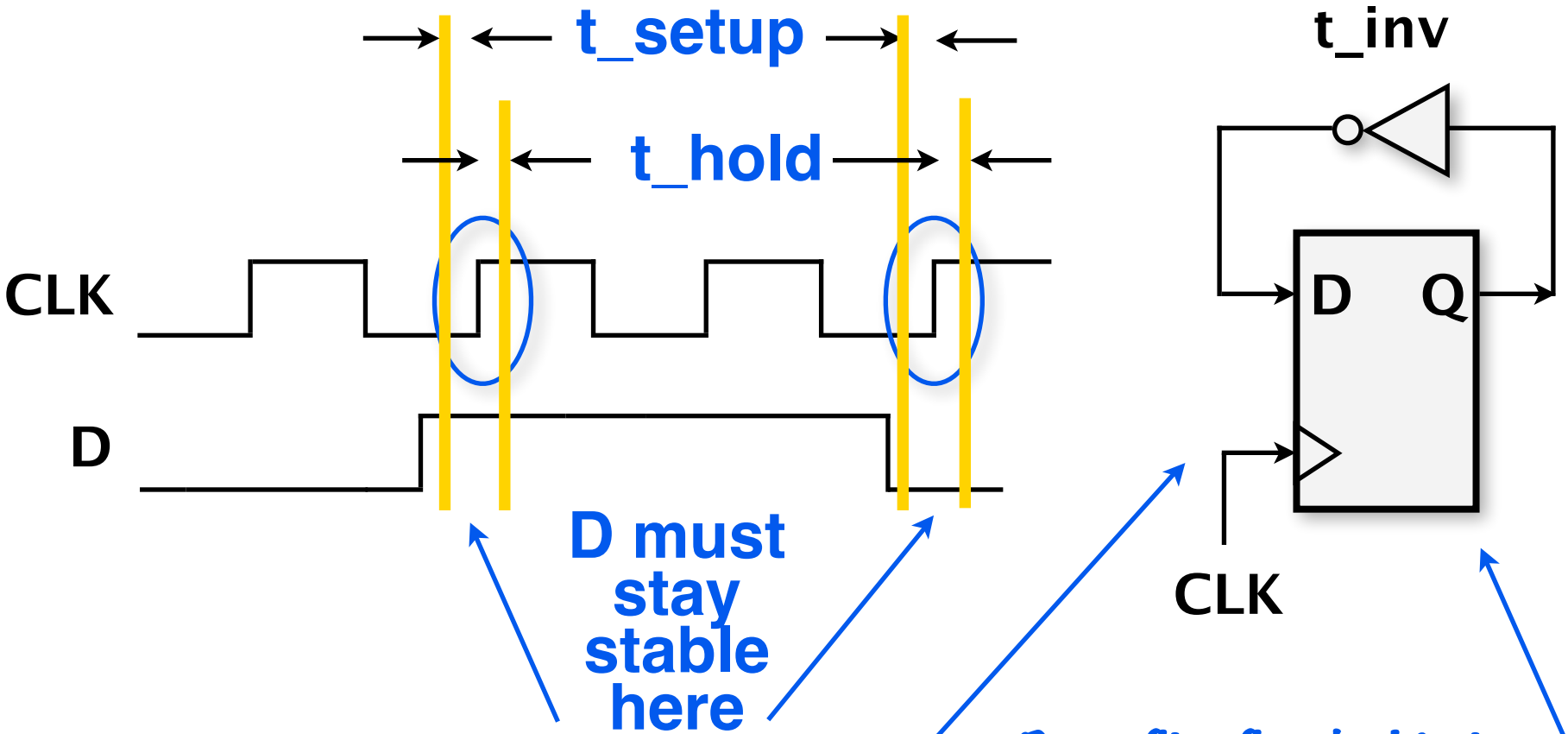
UC Regents Fall 2013/2014 © UCB


Grid
Tuned sector trees
Sector buffers
Delay
Clock Tree Delays, IBM "Power" CPU
Buffer level 2
Buffer level 1
x
y



Clock Tree Delays, IBM Power

Some Flip Flops have "hold" time ...

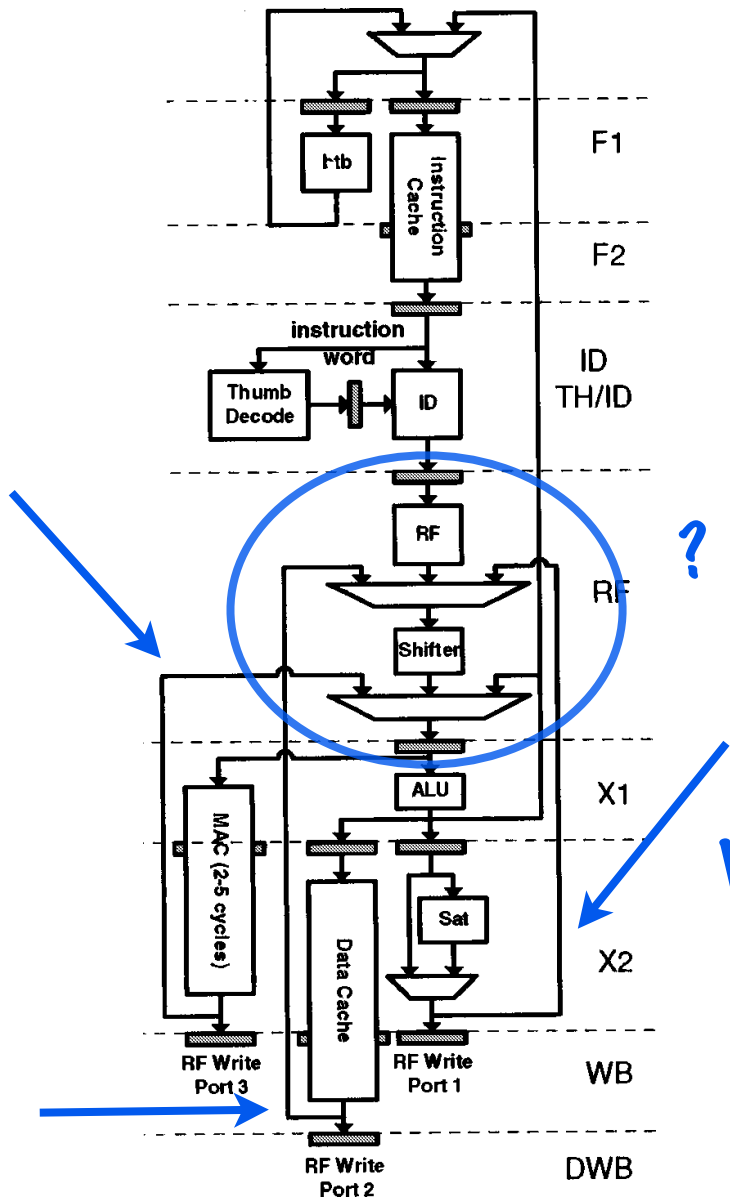


$$t_{clk\text{-}to\text{-}Q} + t_{inv} > t_{hold}$$

For correct operation.

UC Regents Fall 2013/2014 © UCB

Searching for processor critical path



Timing Analysis

What is the smallest T that produces correct operation?

Must consider all connected register pairs.

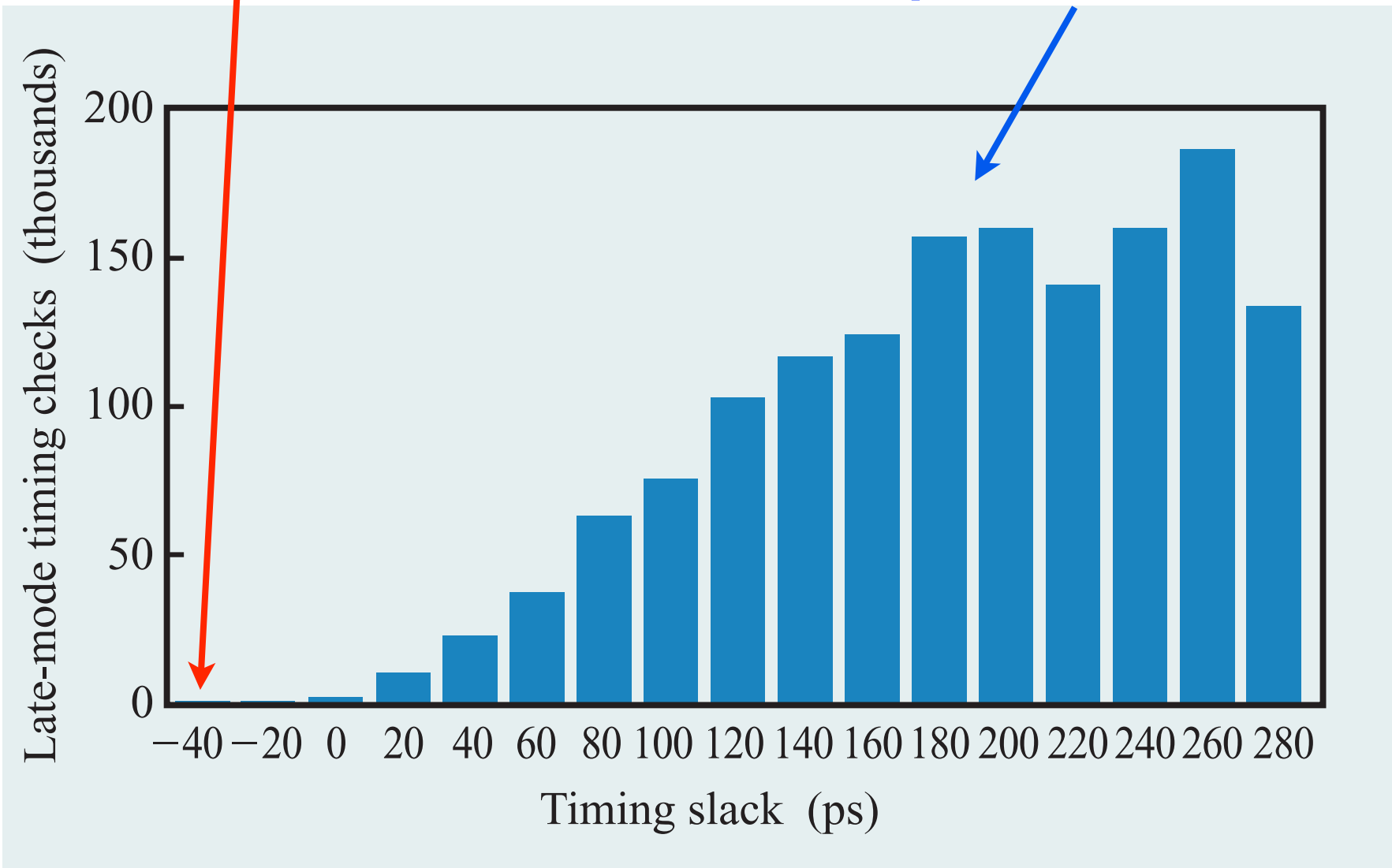
Why might I suspect this one?

UC Regents Fall 2013/2014 © UCB

Combinational paths for IBM Power 4 CPU

The critical path

Most wires have hundreds of picoseconds to spare.



From "The circuit and physical design of the POWER4 microprocessor", IBM J Res and Dev, 46:1, Jan 2002, J.D. Warnock et al.

UC Regents Fall 2013/2014 © UCB

How to retime logic

Circles are combinational logic, labelled with delays.

Critical path is 5.
We want to improve it without changing circuit semantics.

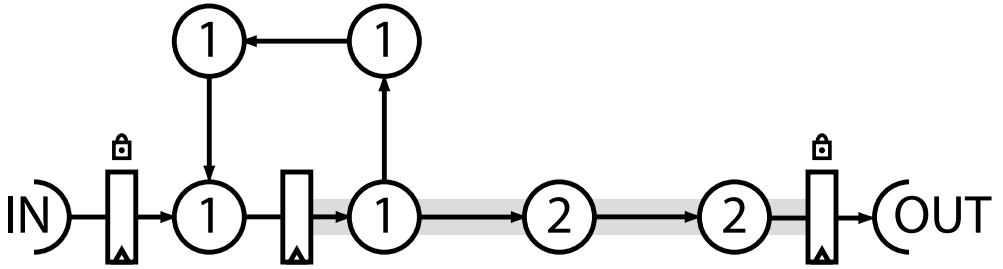


Figure 1: A small graph before retiming. The nodes represent logic delays, with the inputs and outputs passing through mandatory, fixed registers. The critical path is 5.

Add a register, move one circle.
Performance improves by 20%.

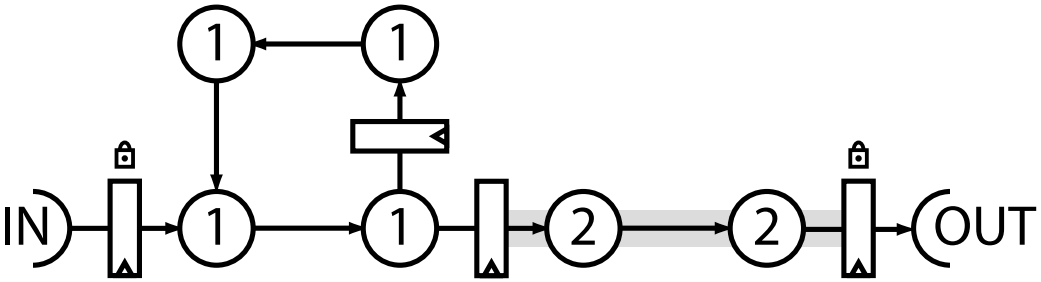


Figure 2: The example in Figure 2 after retiming. The critical path is reduced from 5 to 4.

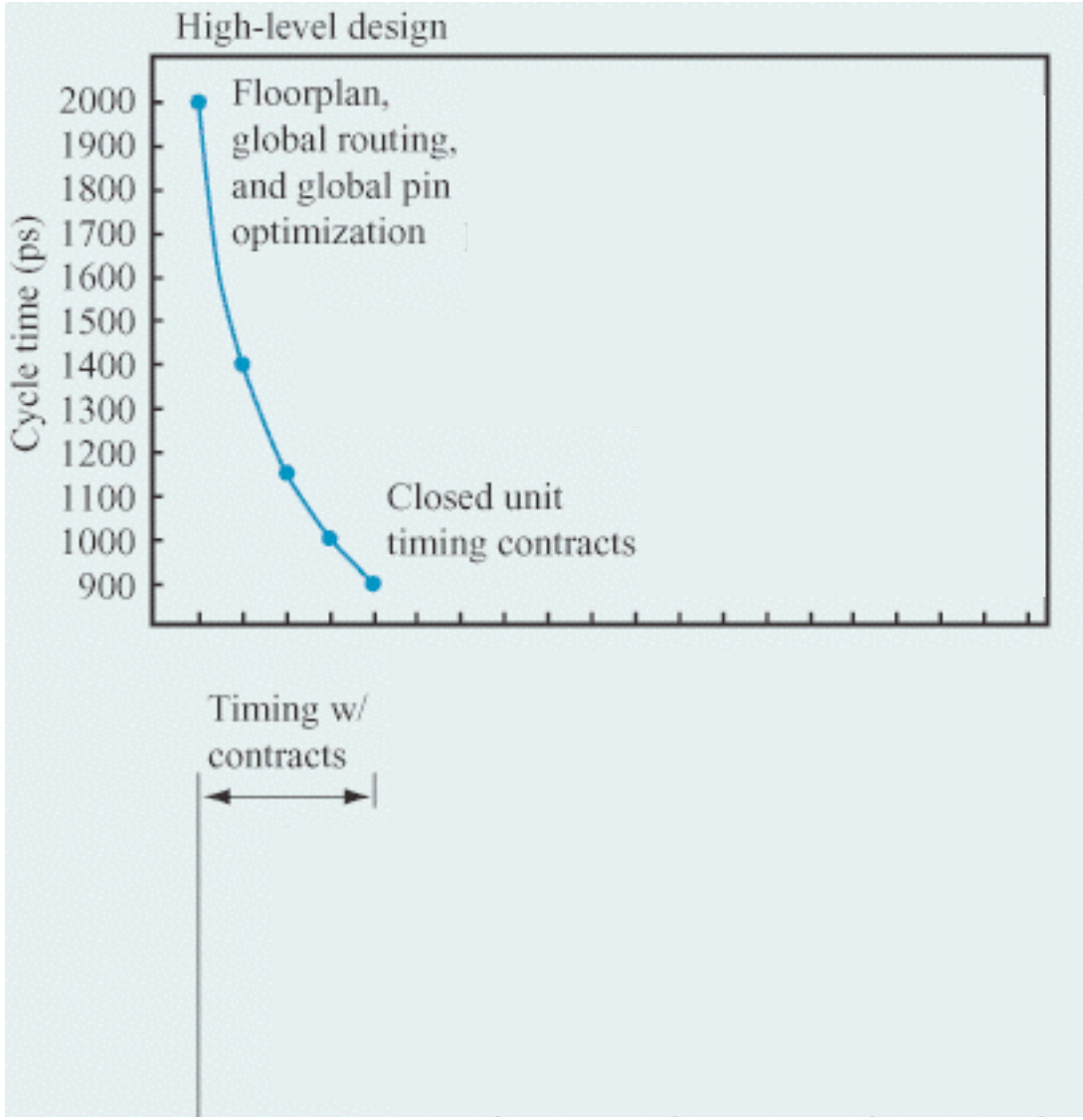
"Technology X" can often do this.

Nicholas Weaver* UC Berkeley Berkeley, CA
Yury Markovskiy UC Berkeley Berkeley, CA
Yatish Patel UC Berkeley Berkeley, CA
John Wawrzynek UC Berkeley Berkeley, CA

Power 4: Timing Estimation, Closure

Timing Estimation

Predicting a processor's clock rate early in the project



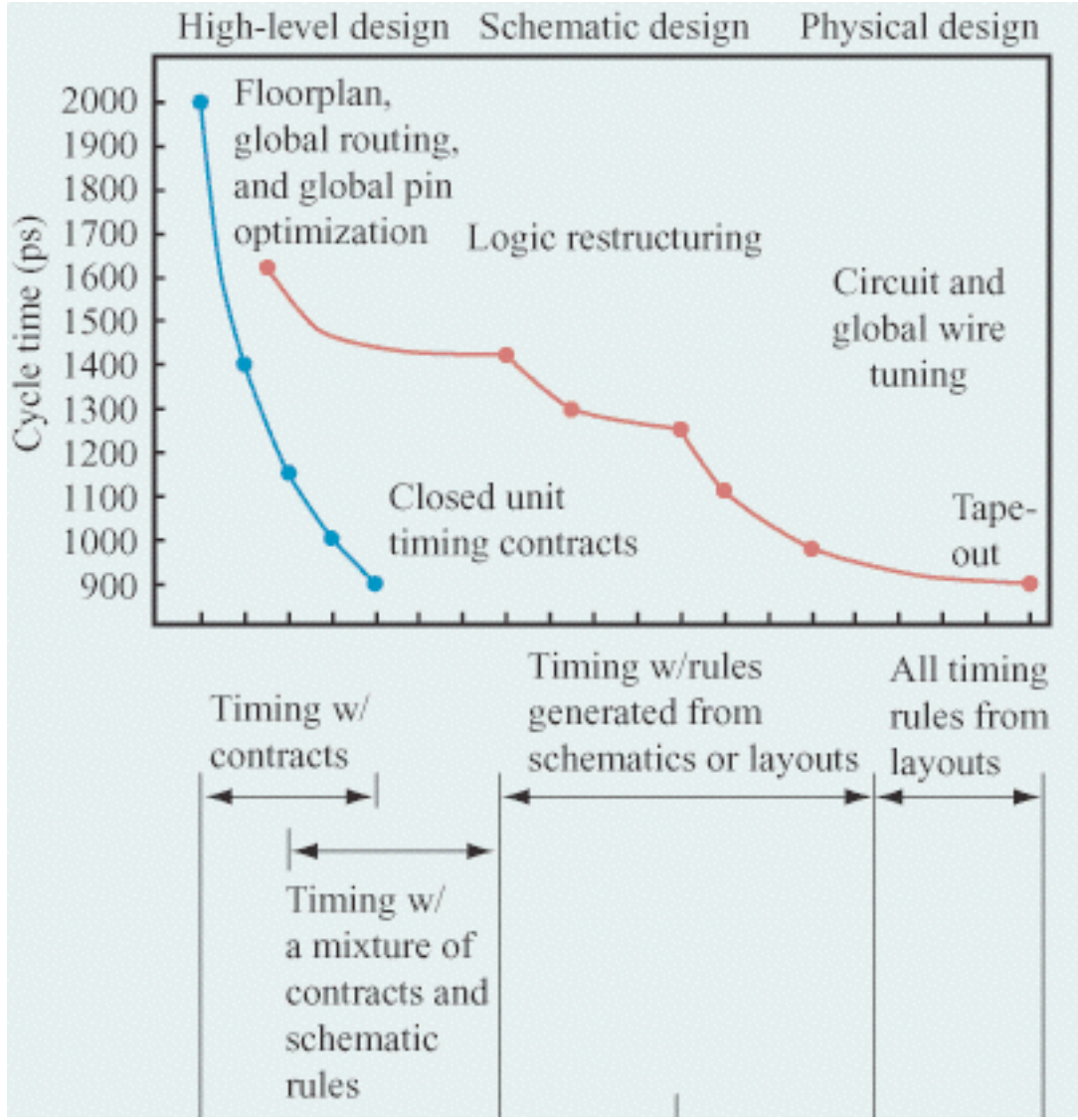
From "The circuit and physical design of the POWER4 microprocessor", IBM J Res and Dev, 46:1, Jan 2002, J.D. Warnock et al.

UC Regents Fall 2013/2014 © UCB

Power 4: Timing Estimation, Closure

Timing Closure

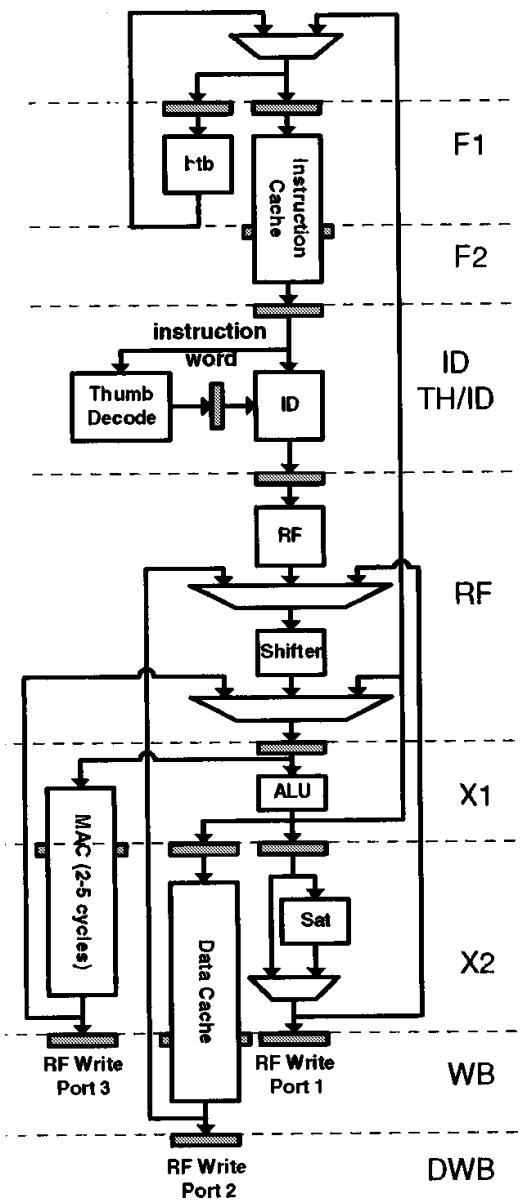
Meeting (or exceeding!) the timing estimate

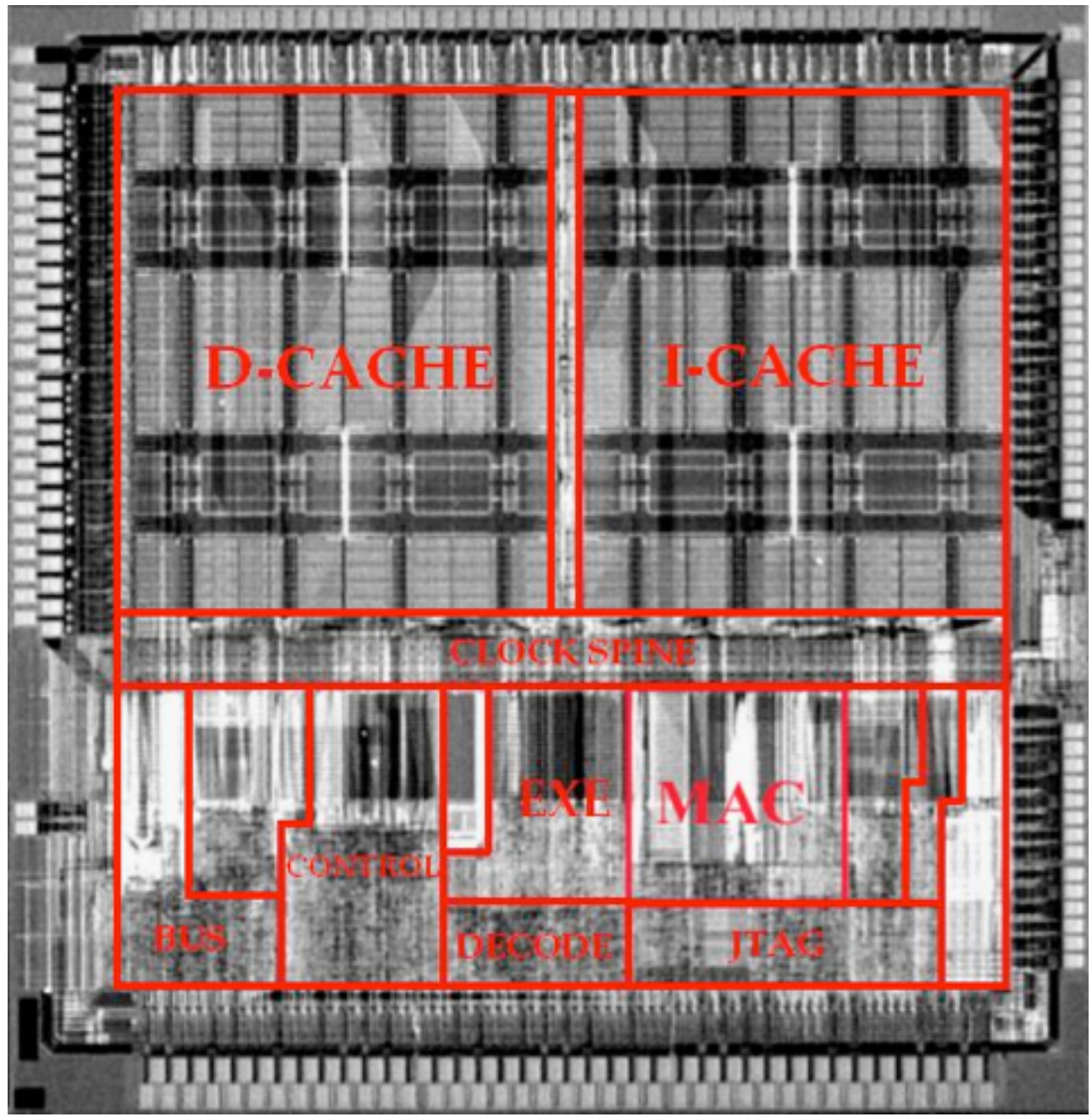


From "The circuit and physical design of the POWER4 microprocessor", IBM J Res and Dev, 46:1, Jan 2002, J.D. Warnock et al.

UC Regents Fall 2013/2014 © UCB

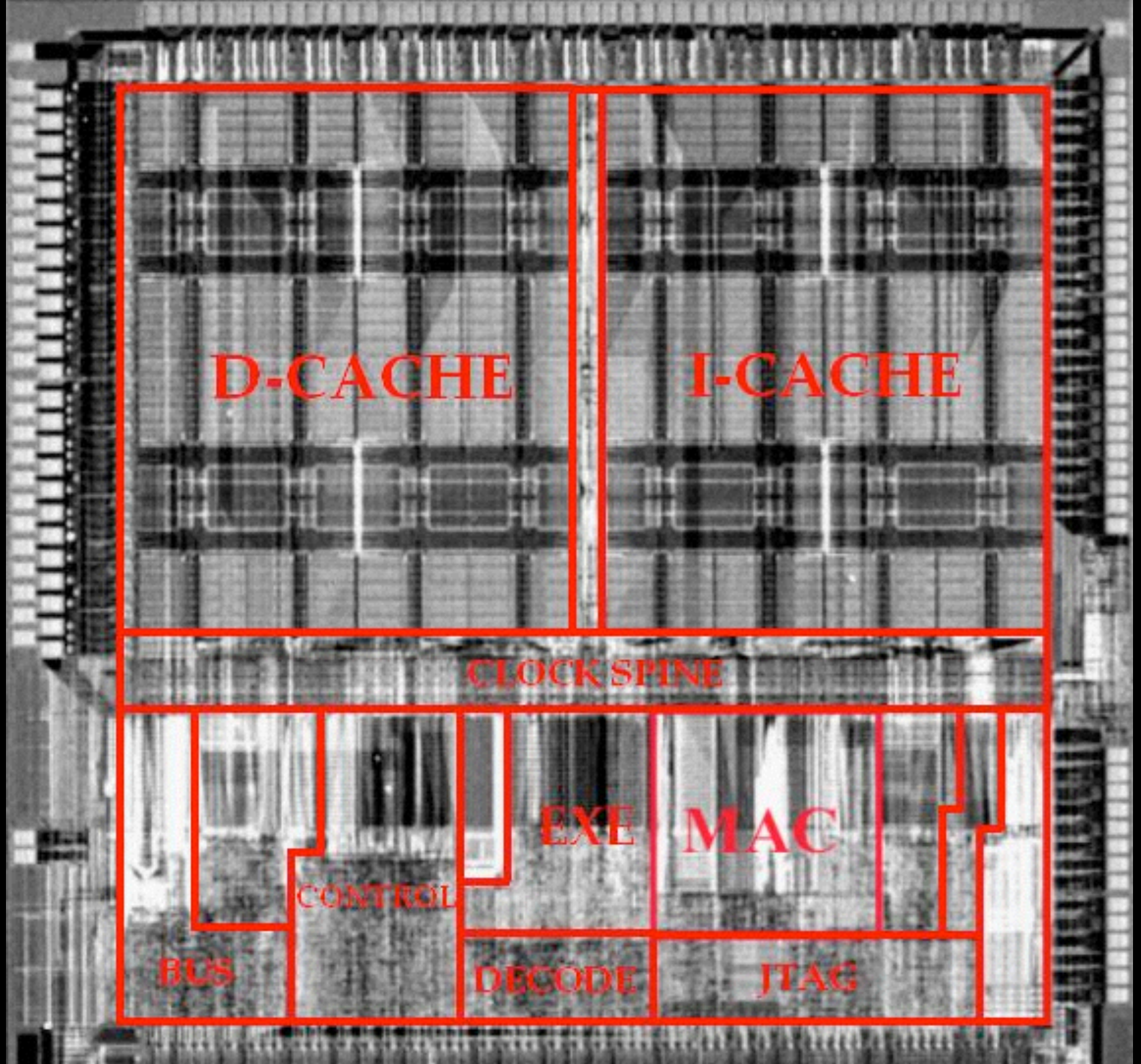
Floorplaning: essential to meet timing.





(Intel XScale 80200)

UC Regents Fall 2013/2014 © UCB


D-CACHE
I-CACHE
CLOCK SPINE
EXE
MAC
CONTROL
BUS
DECODE
JTAG

Break

UC Regents Fall 2013/1014 © UCB

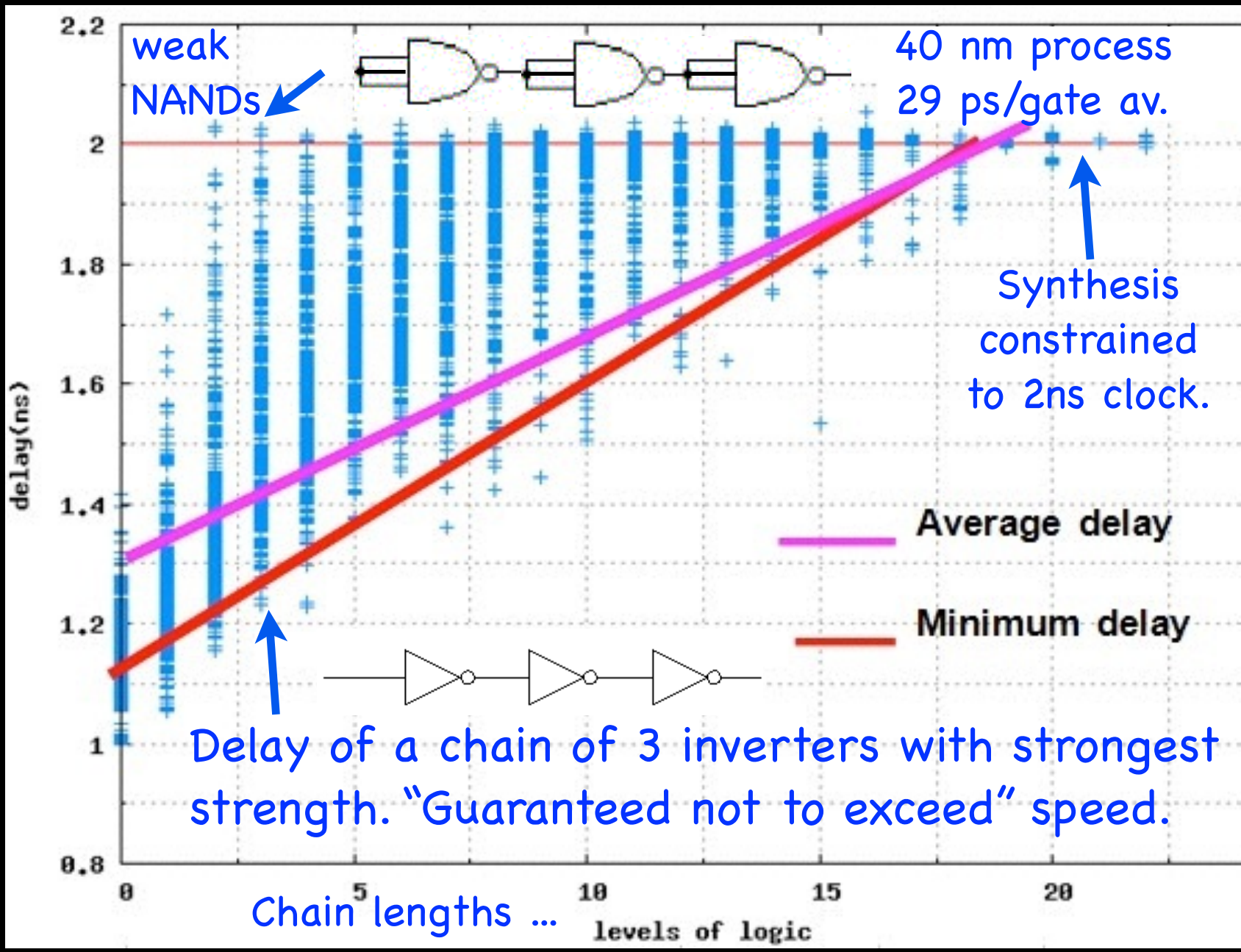
Simple exercises for gaining intuition about timing for your process + EDA tools.

Thanks to Bhupesh Dasila, Open-Silicon Bangalore

Synthesize gate chains using hand-specified library cells

Exercises cell library and place and route tools.

Lets you know how many levels of logic you can use in the best case.



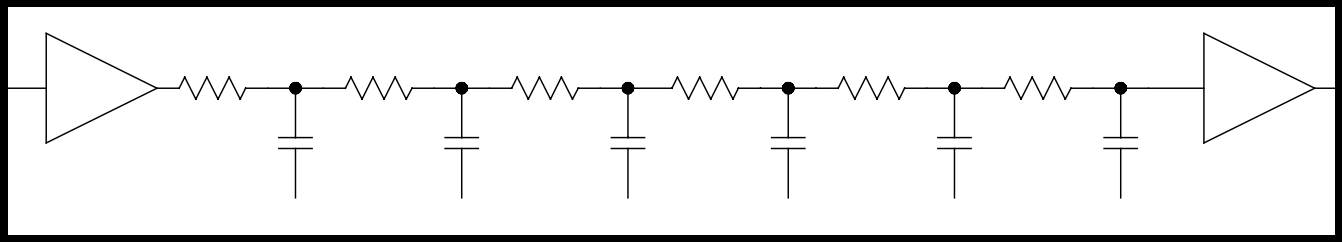
Helps you "see through" ... "Technology X".

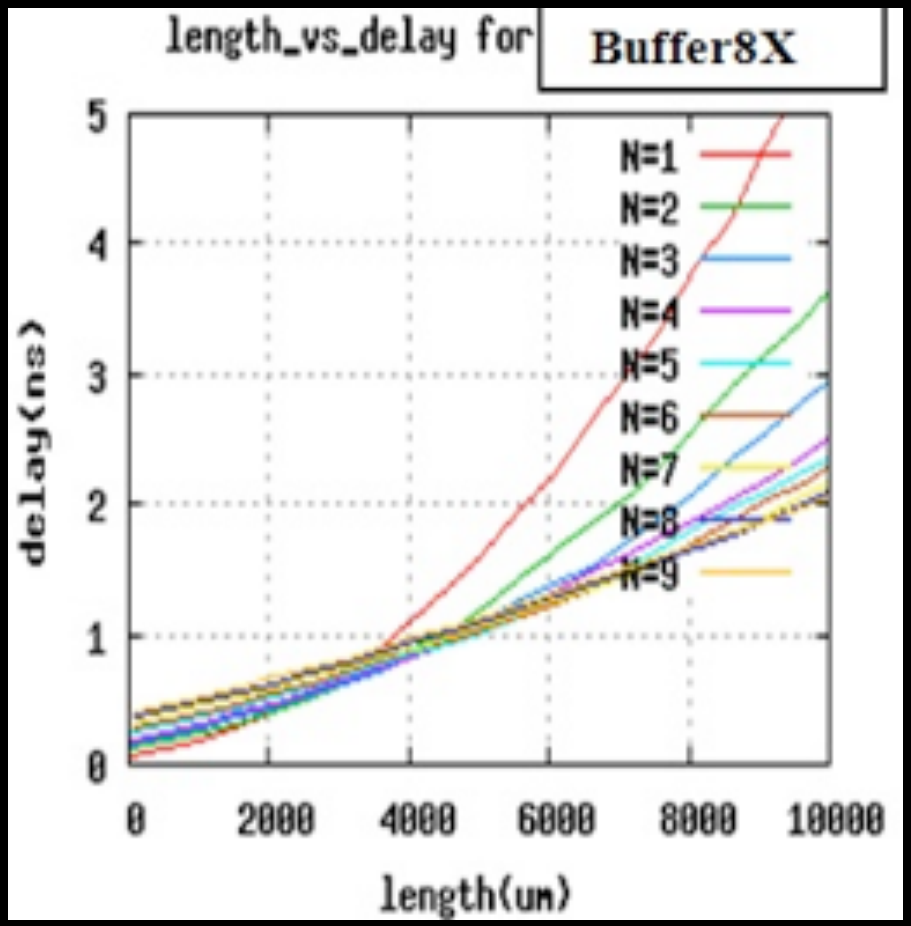
Bhupesh Dasila

Force P&L to drive a long wire with a known buffer cell.

Vary driver strength, wire length, metal layer.

Shows the maximum distance two gates can be placed and still meet your clock period.





Distributed RC is the square of the length is clearly seen!

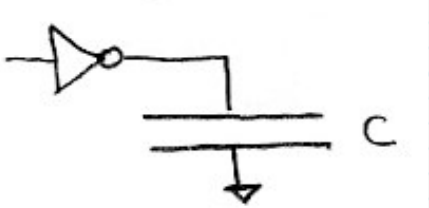
Delay/length = 195.8ps

buffer type	delay/length (ps/mm)	area (um^2)	leakagePower (nW)
Buffer8X	195.8	2.8	1565.6
Buffer12X	175.7	3.9	2199.0
Buffer16X	165.2	5.1	3037.7
Buffer20X	162.0	6.2	3693.8

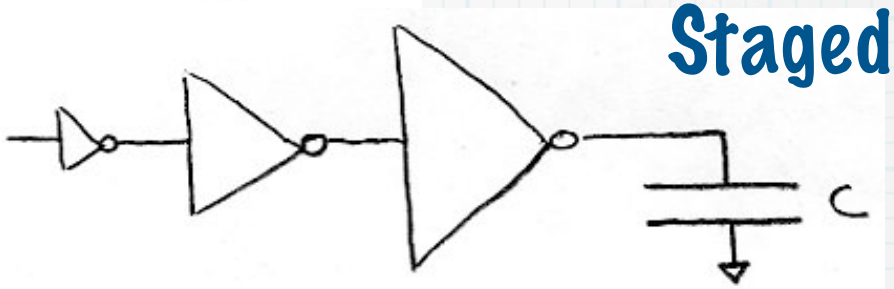
Bhupesh Dasila

Driving Large Loads

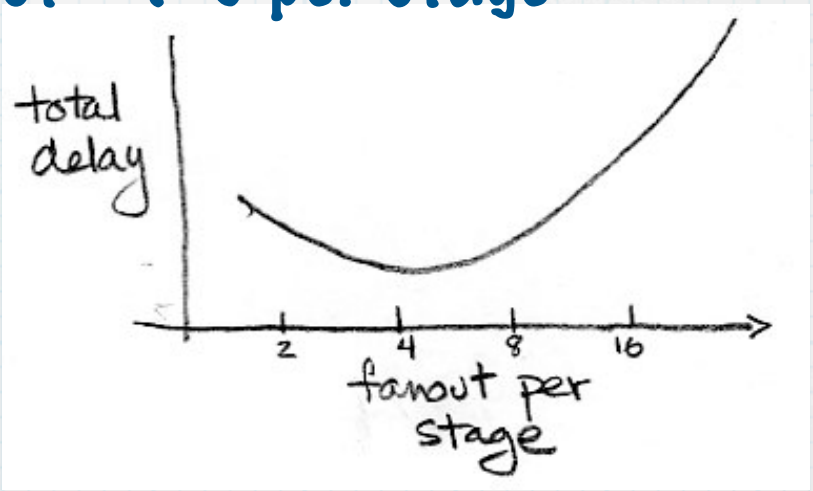
- Large fanout nets: clocks, resets, memory bit lines, off-chip
- Relatively small driver results in long rise time (and thus large gate delay)



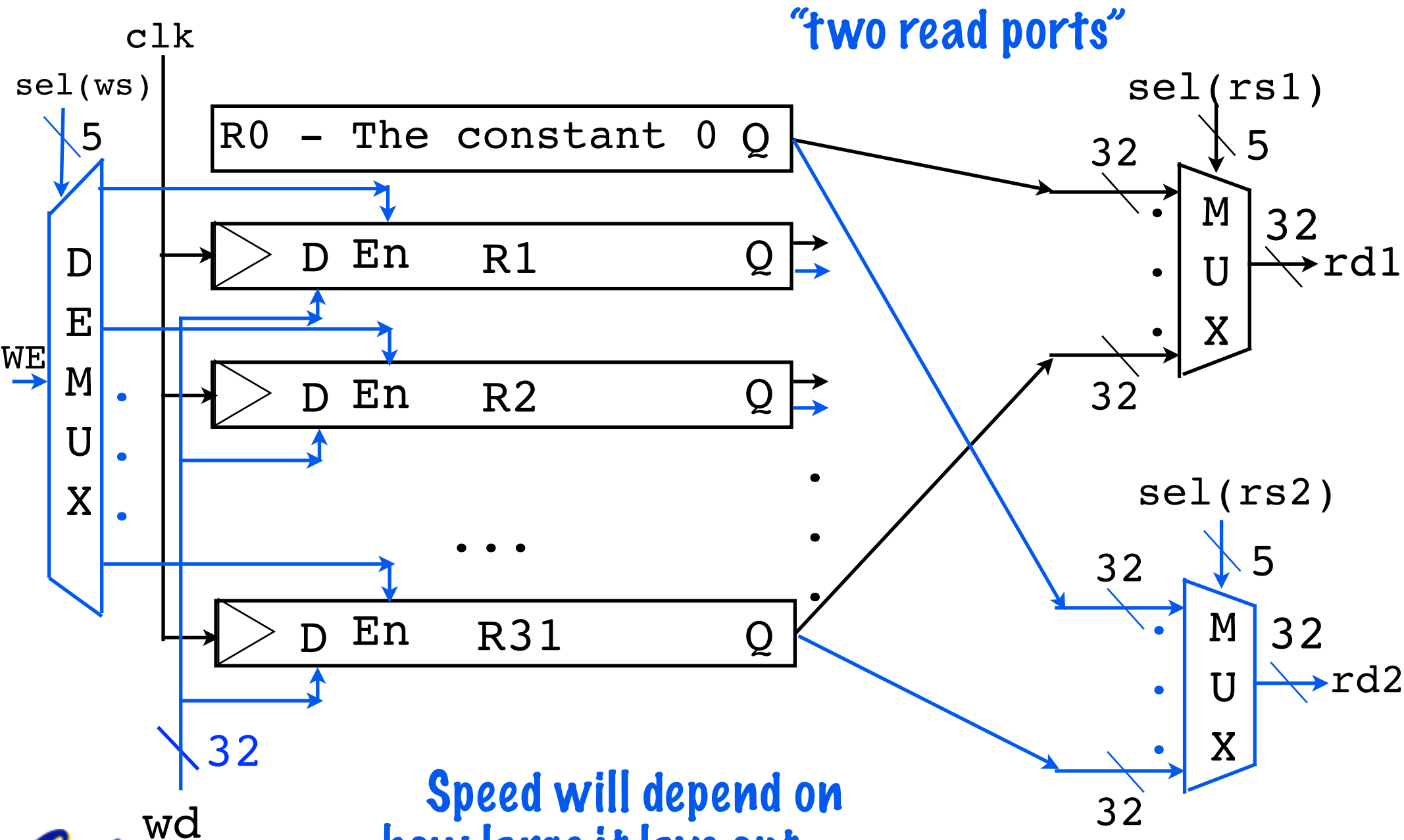
- Strategy: **Staged Buffers**



- Optimal trade-off between delay per stage and total number of stages ⇒ **fanout of ~4-6 per stage**



Register file: Synthesize, or use SRAM?



UC Regents Fall 2013/2014 © UCB

Synthesized, custom, and SRAM-based register files, 40nm

For small register files, logic synthesis is competitive.

Not clear if the SRAM data points include area for register control, etc.

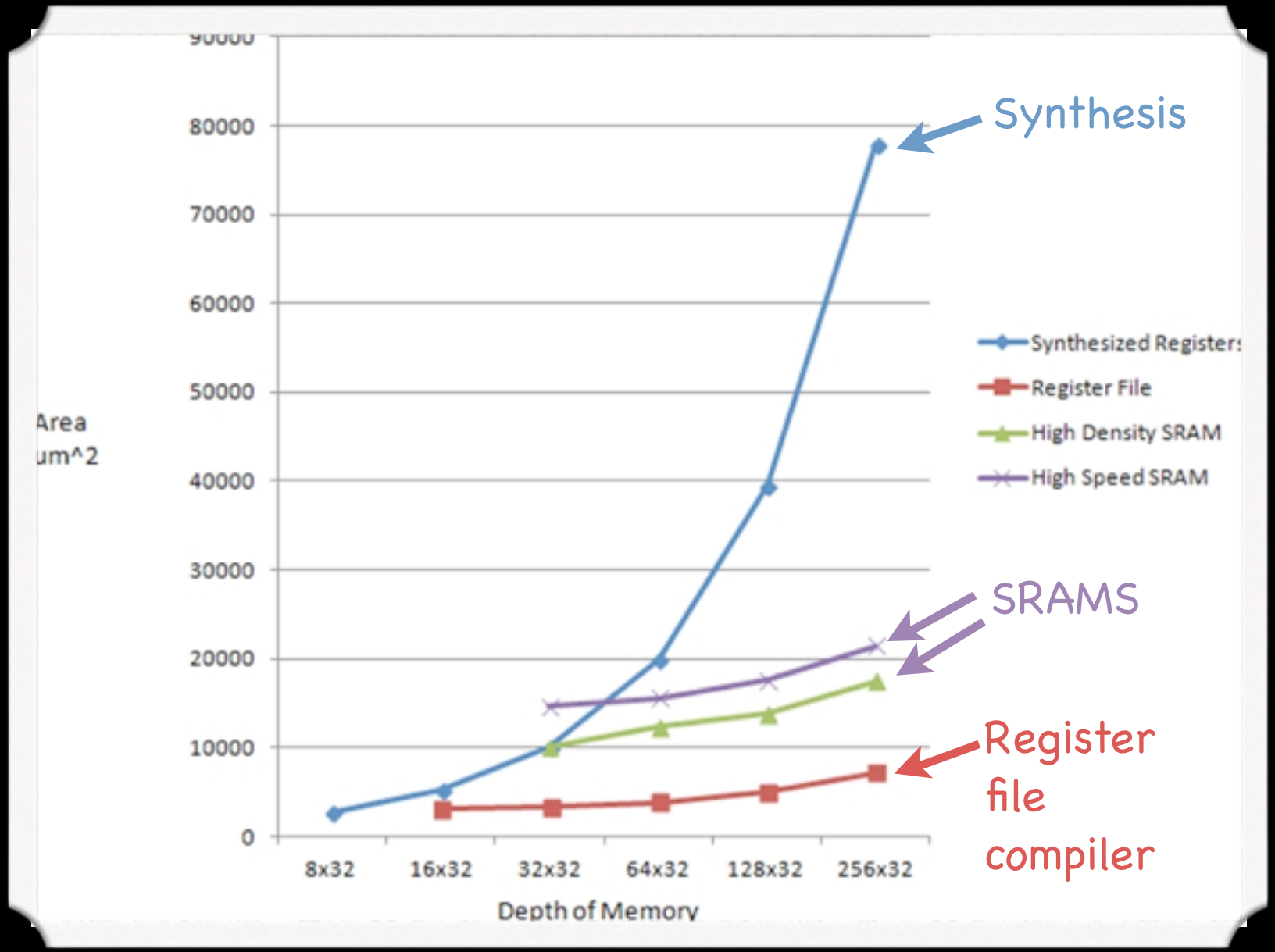


Figure 3: Using the raw area data, the physical implementation team can get a more accurate area estimation early in the RTL development stage for floorplanning purposes. This shows an example of this graph for a 1-port, 32-bit-wide SRAM.

Bhupesh Dasila

Techniques

UC Regents Fall 2013/1014 © UCB

Pipelining

UC Regents Fall 2013 © UCB

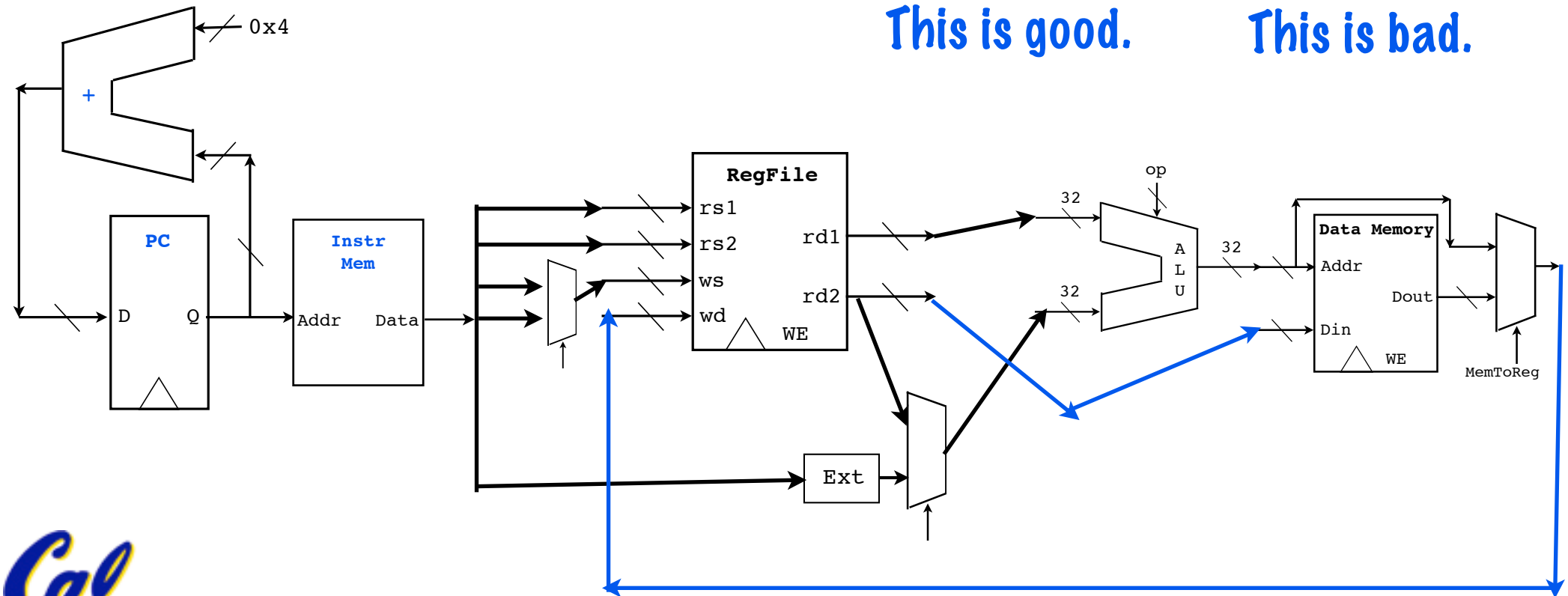
Starting point: A single-cycle processor

Challenge: Speed up clock while keeping CPI == 1

$$\frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \quad \frac{\text{Cycles}}{\text{Instruction}} \quad \frac{\text{Seconds}}{\text{Cycle}}$$

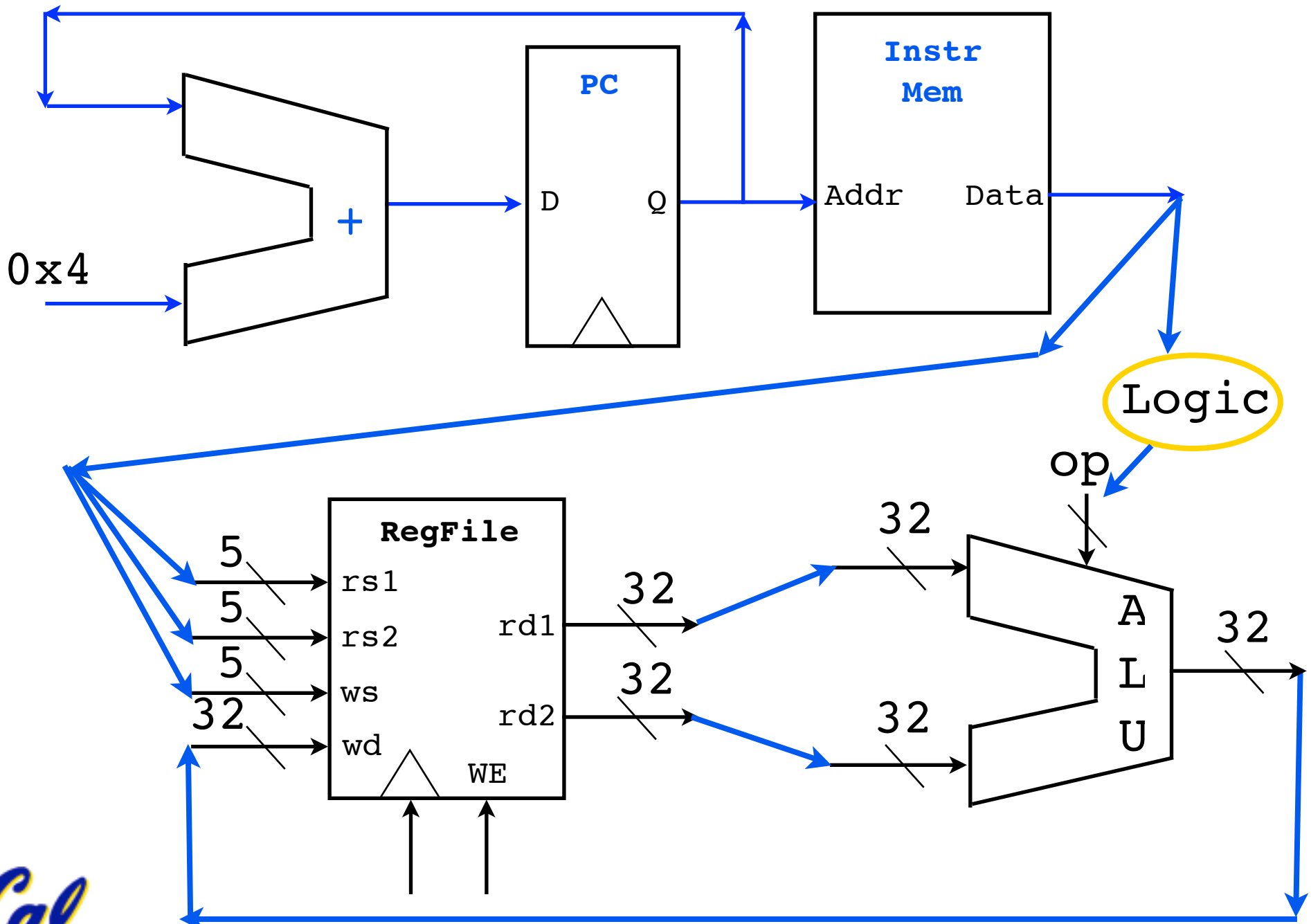
CPI == 1
This is good.

Slow.
This is bad.



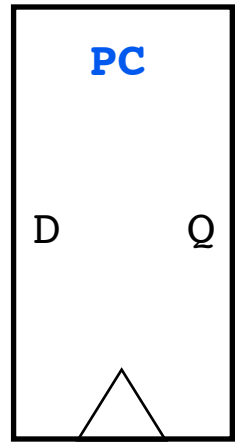
UC Regents Fall 2013/2014 © UCB

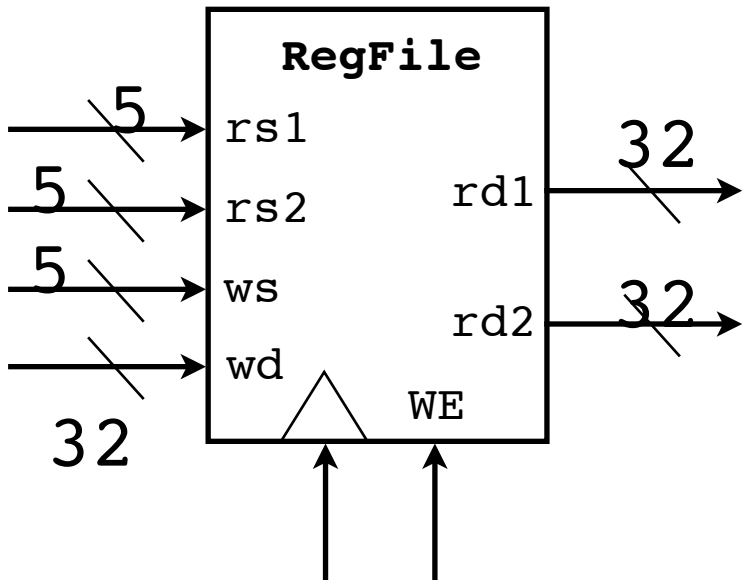
Reminder: How data flows after posedge



UC Regents Fall 2013/2014 © UCB

Next posedge: Update state and repeat



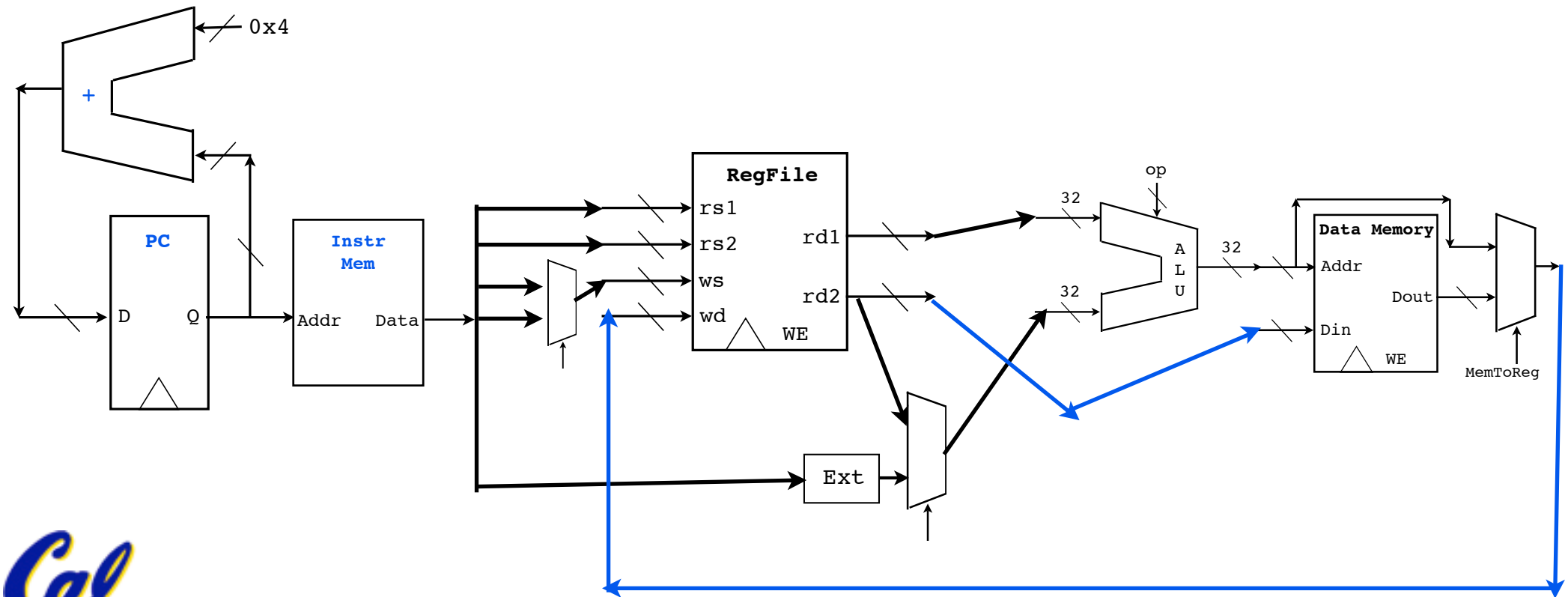


UC Regents Fall 2013/2014 © UCB

Observation: Logic idle most of cycle

For most of cycle, ALU is either "waiting" for its inputs, or "holding" its output

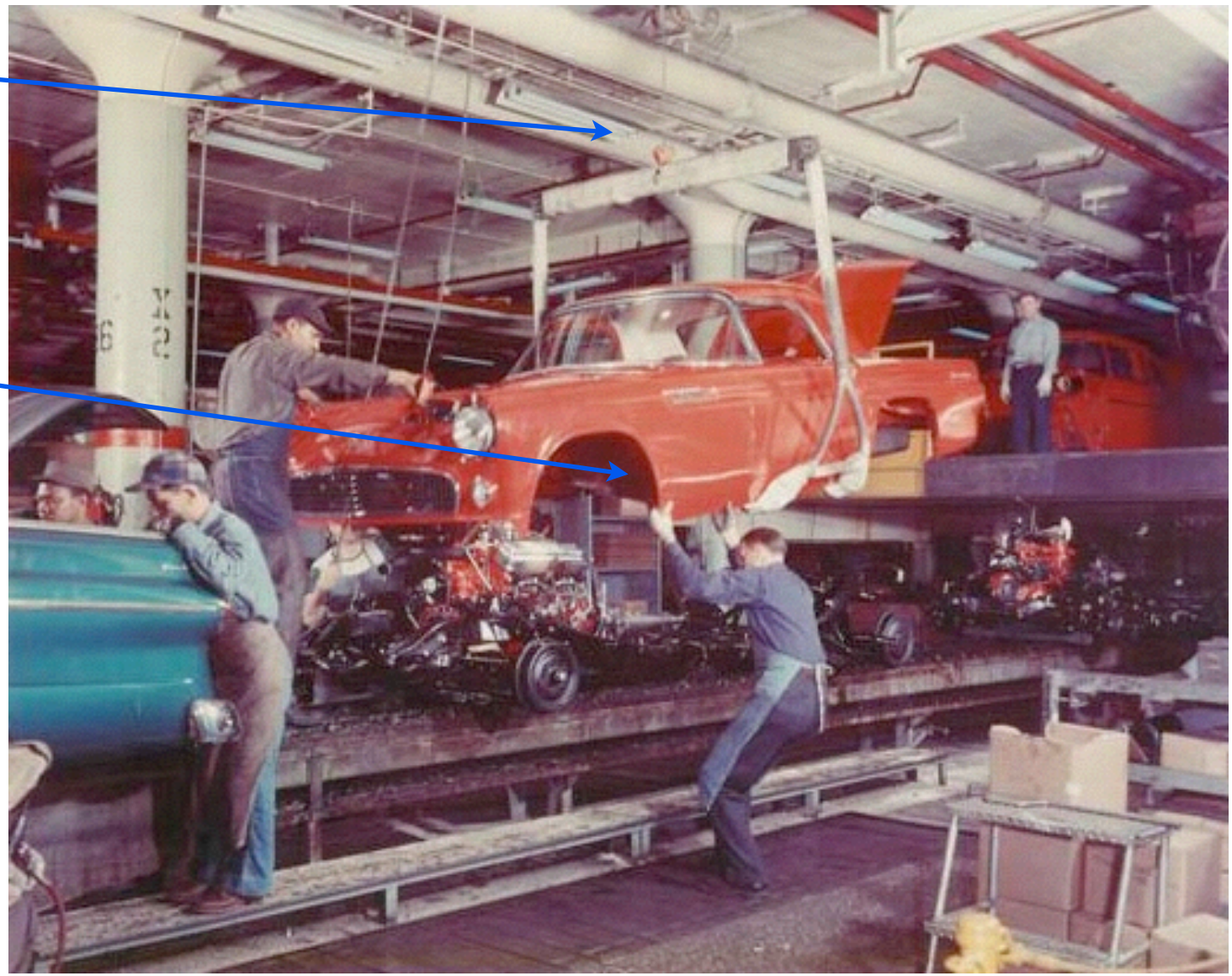
Ideal: a CPU architecture where each part is always "working".



UC Regents Fall 2013/2014 © UCB

Inspiration: Automobile assembly line

Assembly line moves on a steady clock.
Each station does the same task on each car.



UC Regents Fall 2013/2014 © UCB

Inspiration: Automobile assembly line

Simpler station tasks → more cars per hour.
Simple tasks take less time, clock is faster.

UC Regents Fall 2013 © UCB

Inspiration: Automobile assembly line

Line speed limited by slowest task.

Most efficient if all tasks take same time to do

UC Regents Fall 2013 © UCB

Inspiration: Automobile assembly line

Simpler tasks, complex car → long line!

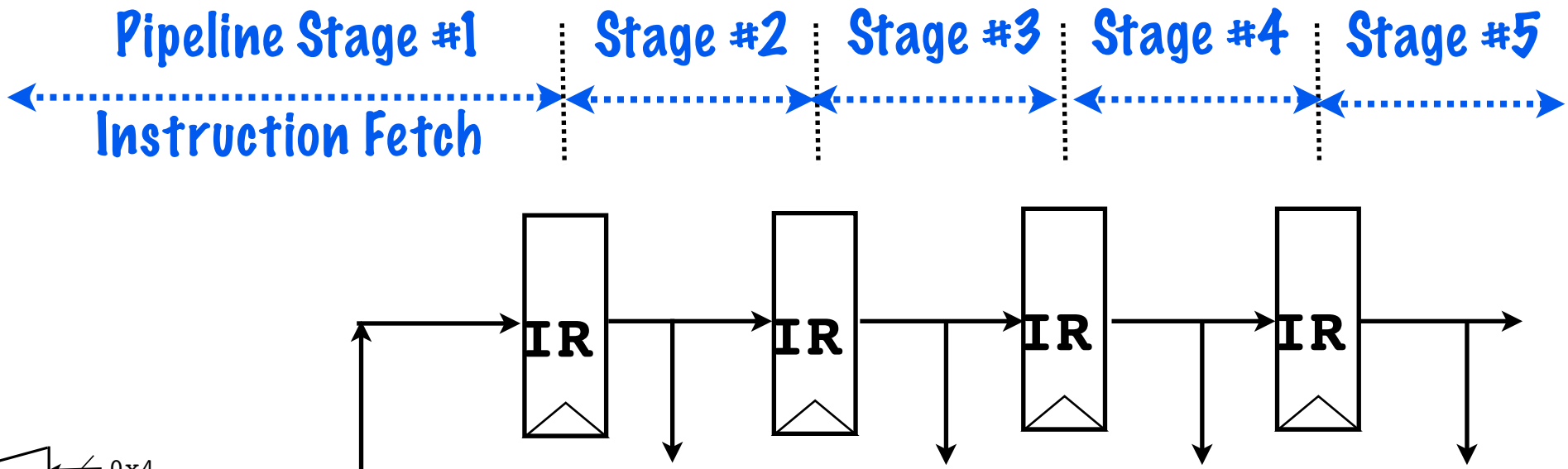
UC Regents Fall 2013 © UCB

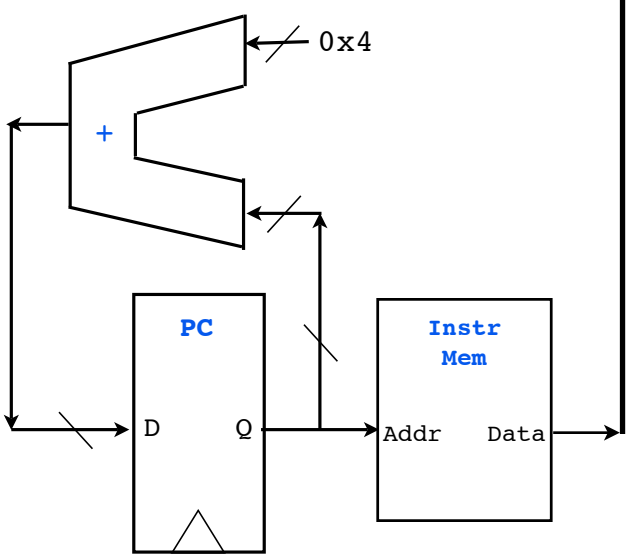
Lessons from car assembly lines

- Faster line movement yields more cars per hour off the line.
- Faster line movement requires more stages, each doing simpler tasks.
- To maximize efficiency, all stages should take same amount of time (if not, workers in fast stages are idle)
- "Filling", "flushing", and "stalling" assembly line are all bad news.

UC Regents Fall 2013/2014 © UCB

Key Analogy: The instruction is the car





Controls hardware in stage 2

Controls hardware in stage 3

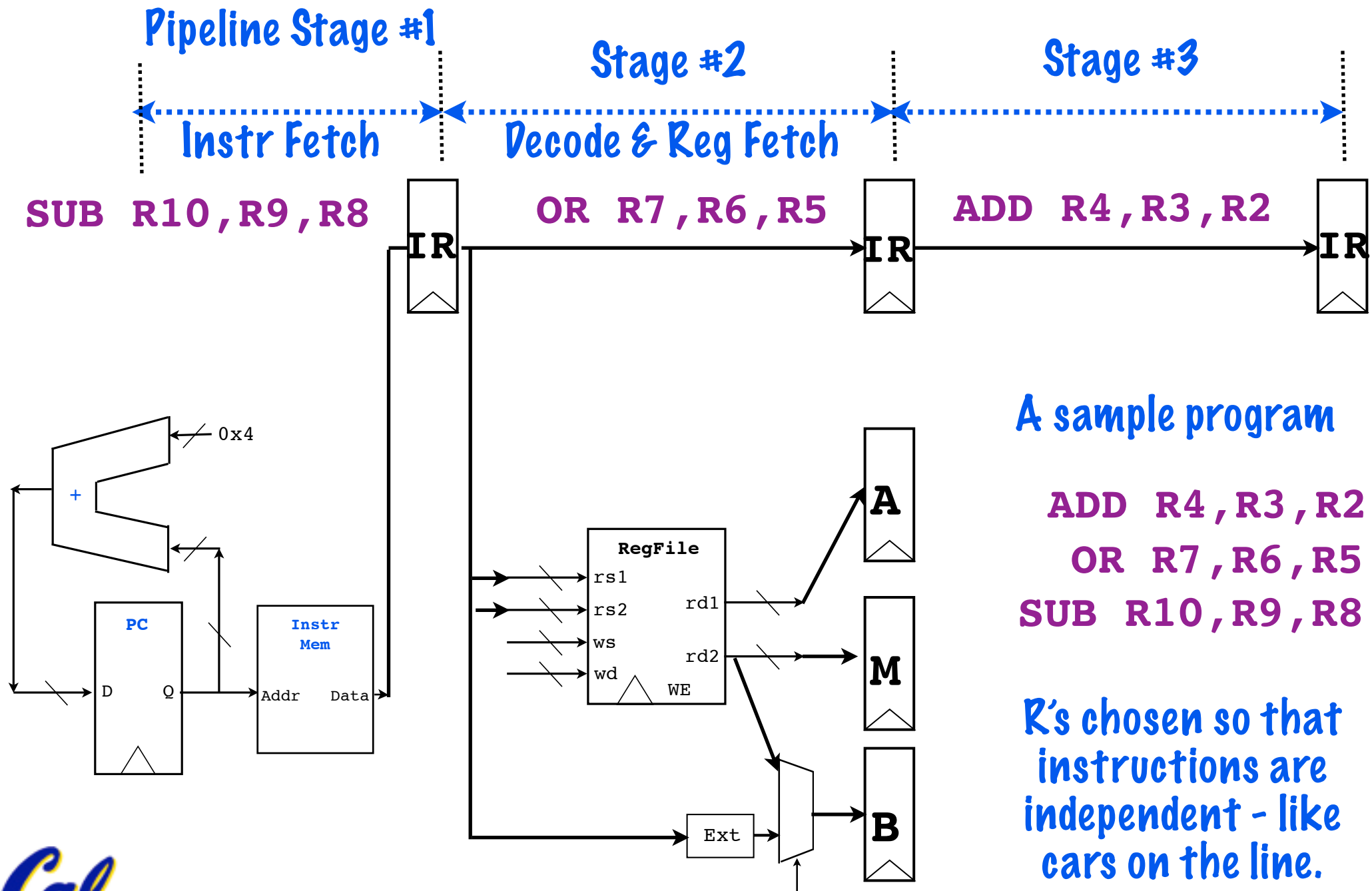
Controls hardware in stage 4

Controls hardware in stage 5

"Data-stationary control"

UC Regents Fall 2013/2014 © UCB

Example: Decode & Register Fetch Stage



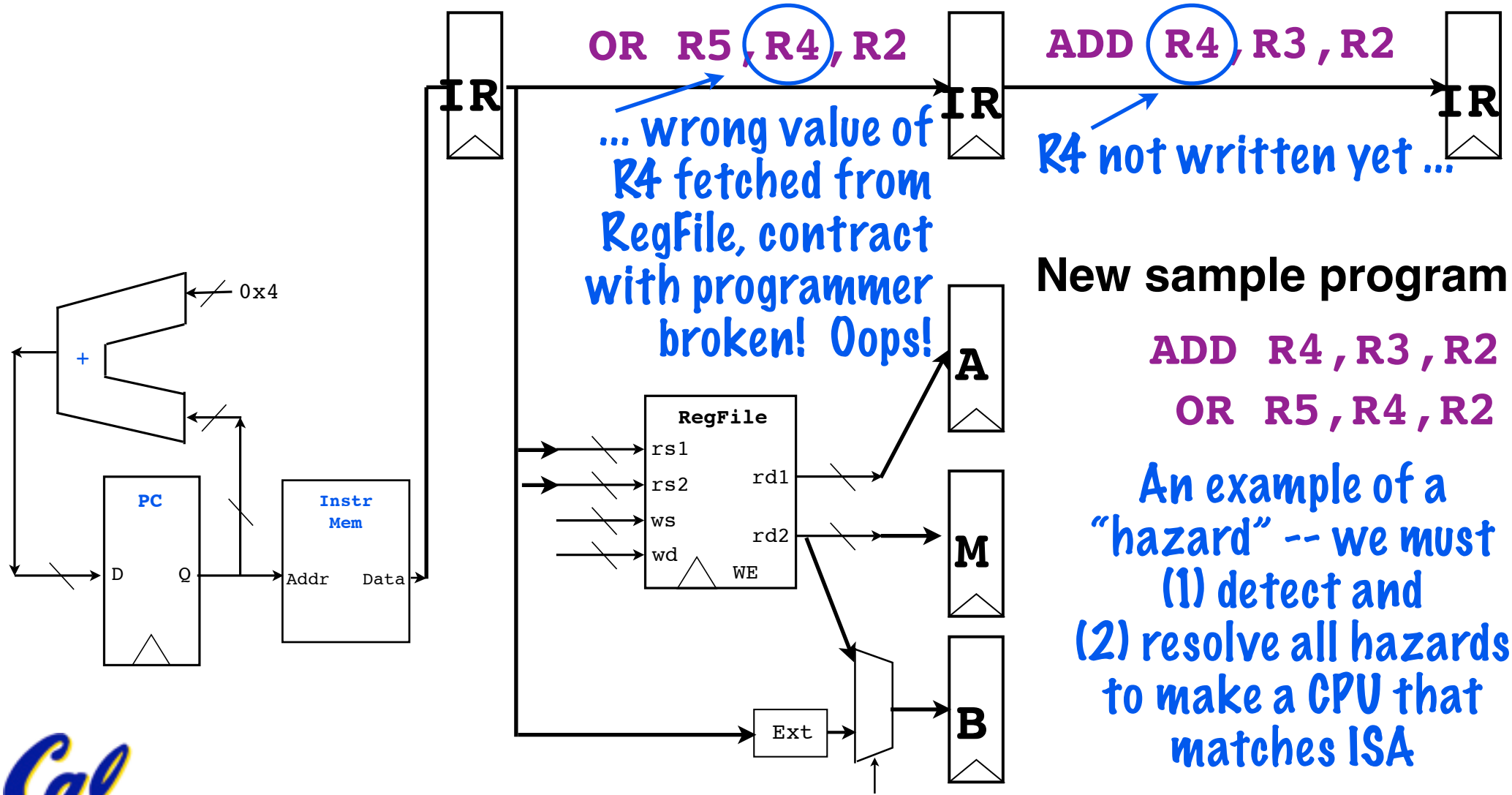
A sample program

```
ADD R4,R3,R2
OR R7,R6,R5
SUB R10,R9,R8
```

R's chosen so that instructions are independent - like cars on the line.

UC Regents Fall 2013/2014 © UCB

Hazards: An instruction is not a car ...



New sample program

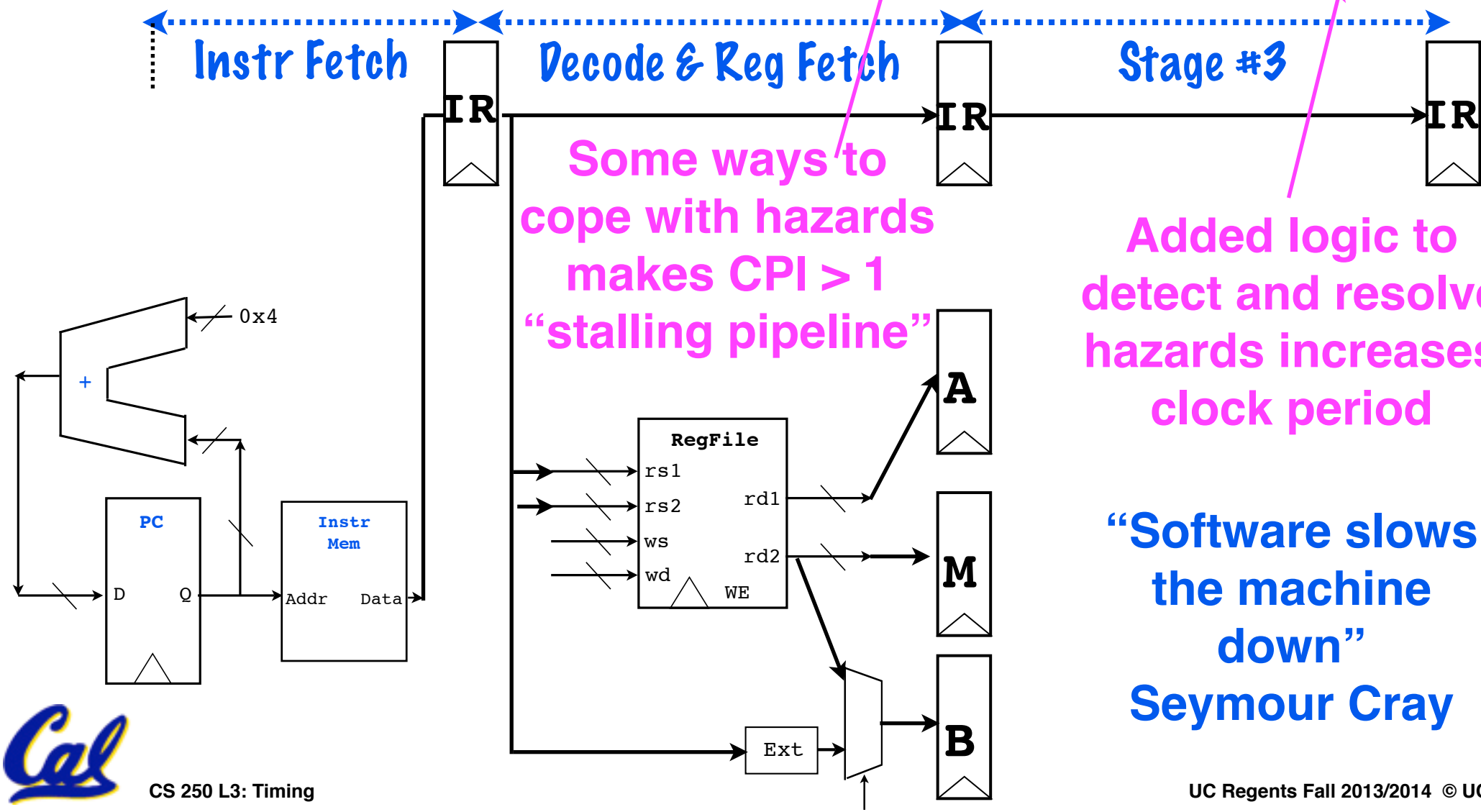
```
ADD R4,R3,R2
OR R5,R4,R2
```

An example of a "hazard" -- we must (1) detect and (2) resolve all hazards to make a CPU that matches ISA

UC Regents Fall 2013/2014 © UCB

Performance Equation and Hazards

$$\frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \cdot \frac{\text{Cycles}}{\text{Instruction}} \cdot \frac{\text{Seconds}}{\text{Cycle}}$$



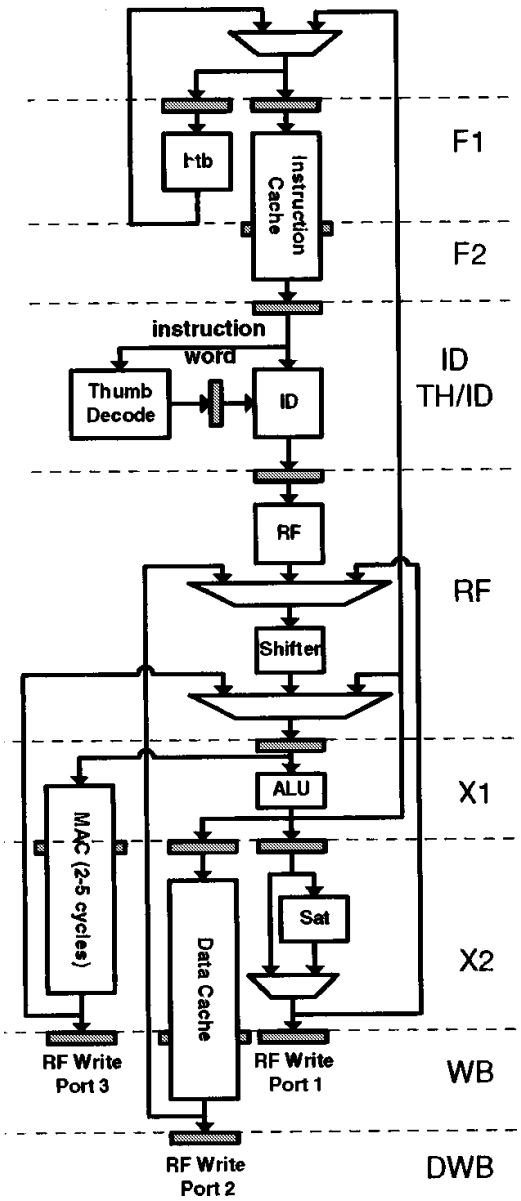
"Software slows the machine down"
Seymour Cray

UC Regents Fall 2013/2014 © UCB

Superpipelining

UC Regents Fall 2013 © UCB

Superpipelining: Add more stages



$$\frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \quad \frac{\text{Cycles}}{\text{Instruction}} \quad \frac{\text{Seconds}}{\text{Cycle}}$$

Goal: Reduce critical path by adding more pipeline stages.

Example: 8-stage ARM XScale: extra IF, ID, data cache stages.

Difficulties: Added penalties for load delays and branch misses.

Ultimate Limiter: As logic delay goes to 0, FF clk-to-Q and setup.

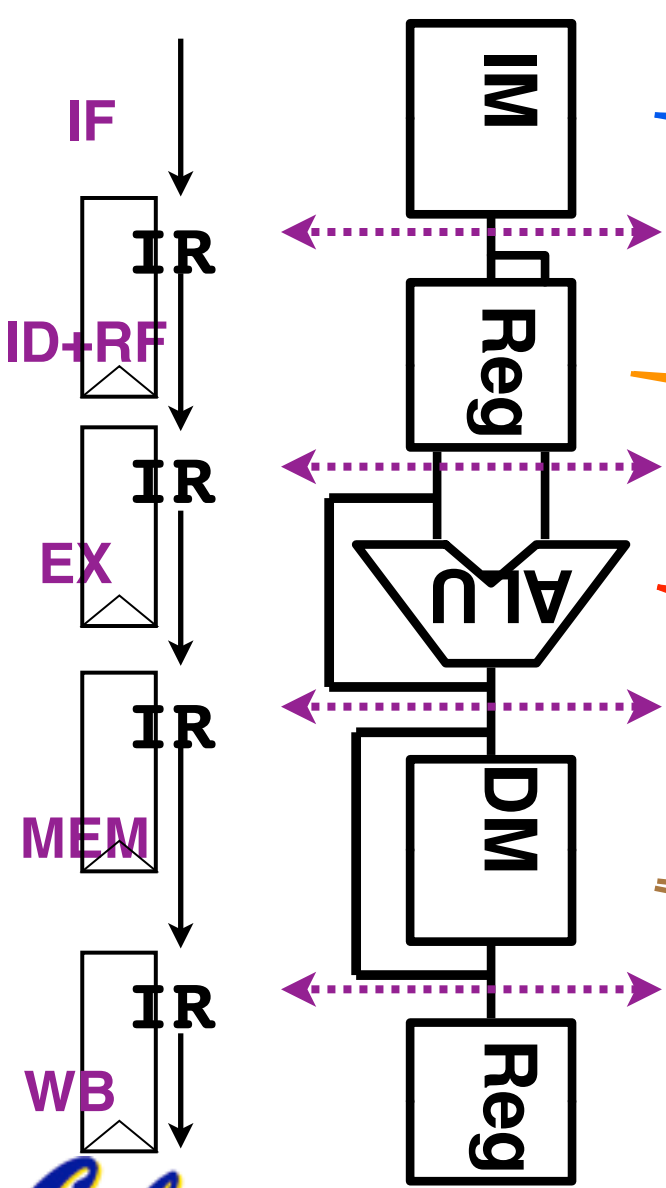
Also, power!

UC Regents Fall 2013/2014 © UCB

5 Stage

Note: Some stages now overlap, some instructions take extra stages.

8 Stage

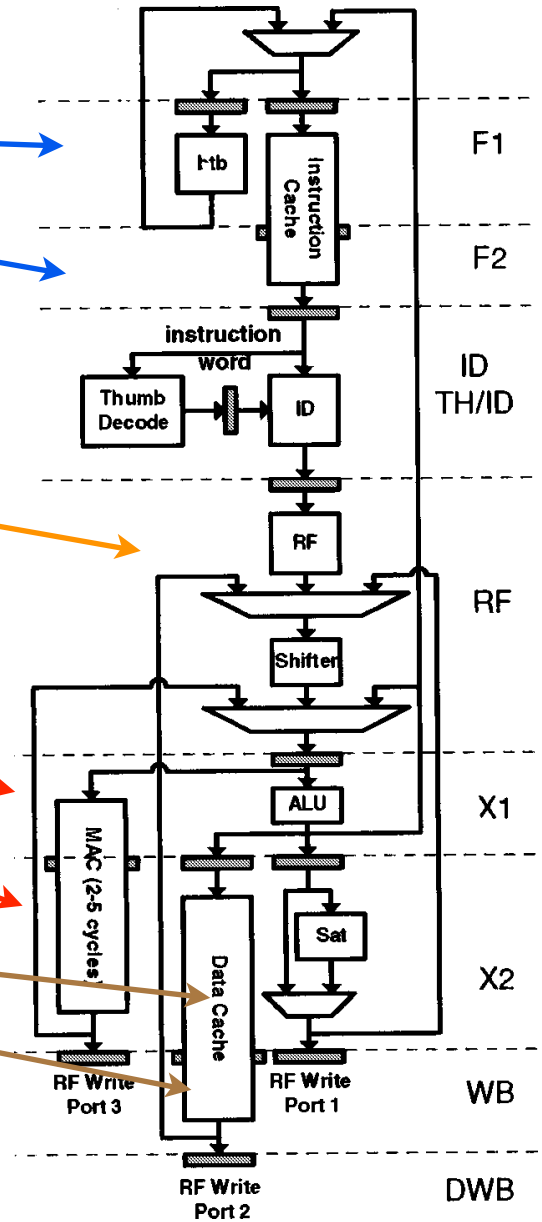


IF now takes 2 stages (pipelined I-cache)

ID and RF each get a stage.

ALU split over 3 stages

MEM takes 2 stages (pipelined D-cache)



UC Regents Fall 2013/1014 © UCB

Superpipelining techniques ...

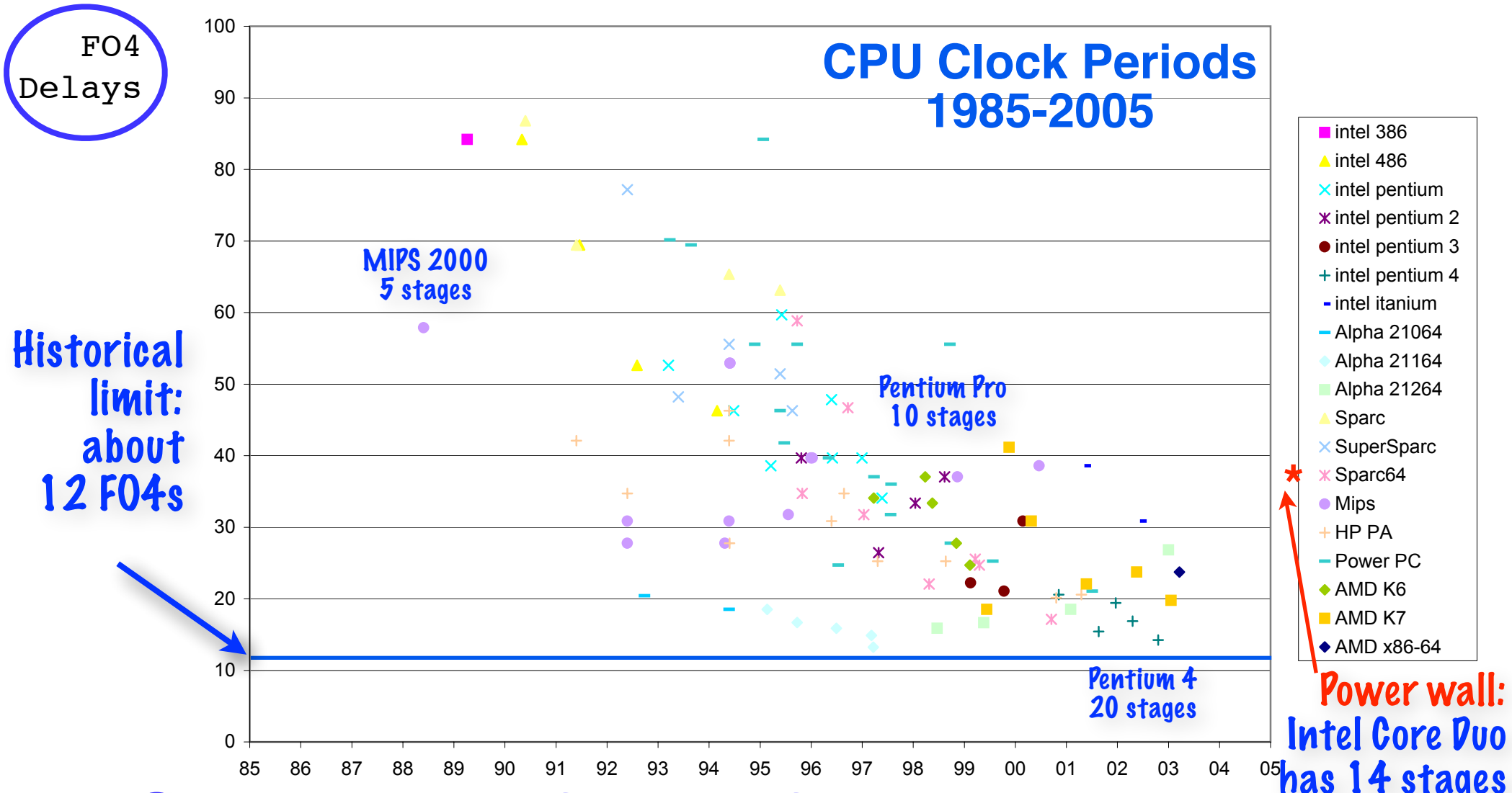
- Split ALU and decode logic over several pipeline stages.
- Pipeline memory: Use more banks of smaller arrays, add pipeline stages between decoders, muxes.
- Remove "rarely-used" forwarding networks that are on critical path.

  Creates stalls, affects CPI.
- Pipeline the wires of frequently used forwarding networks.

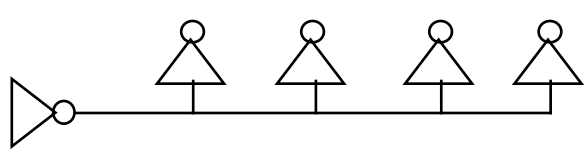
Also: Clocking tricks (example: use posedge and negedge registers)

UC Regents Fall 2013 © UCB

Hardware limits to superpipelining?



FO4: How many fanout-of-4 inverter delays in the clock period.



Thanks to Francois Labonte, Stanford

UC Regents Fall 2013/2014 © UCB

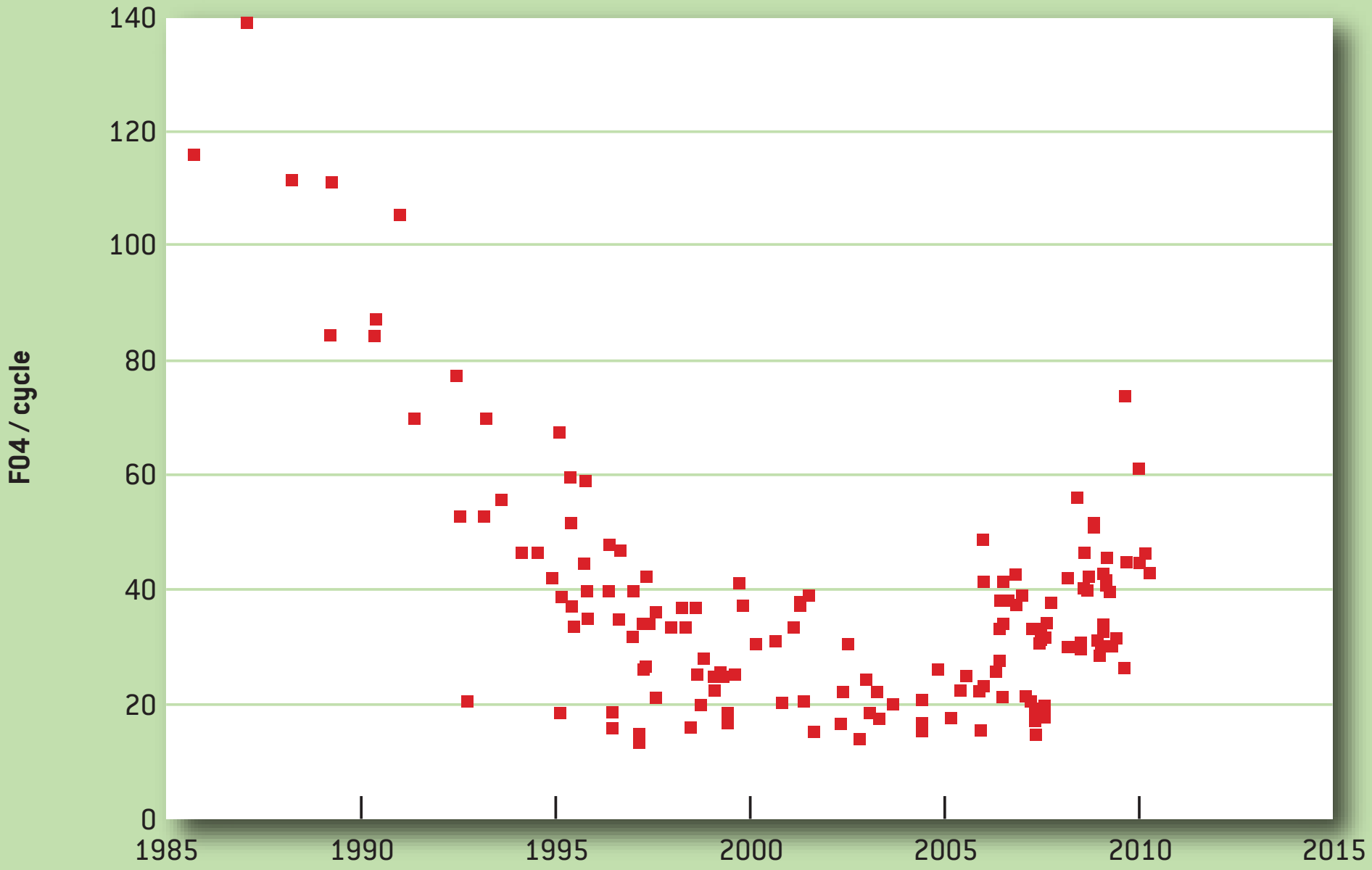
acmqueue

CPU DB: Recording Microprocessor History

With this open database, you can mine microprocessor trends over the past 40 years.

Andrew Danowitz, Kyle Kelley, James Mao, John P. Stevenson, Mark Horowitz, Stanford University

FO4 Delays Per Cycle for Processor Designs



FO4 delay per cycle is roughly proportional to the amount of computation completed per cycle.

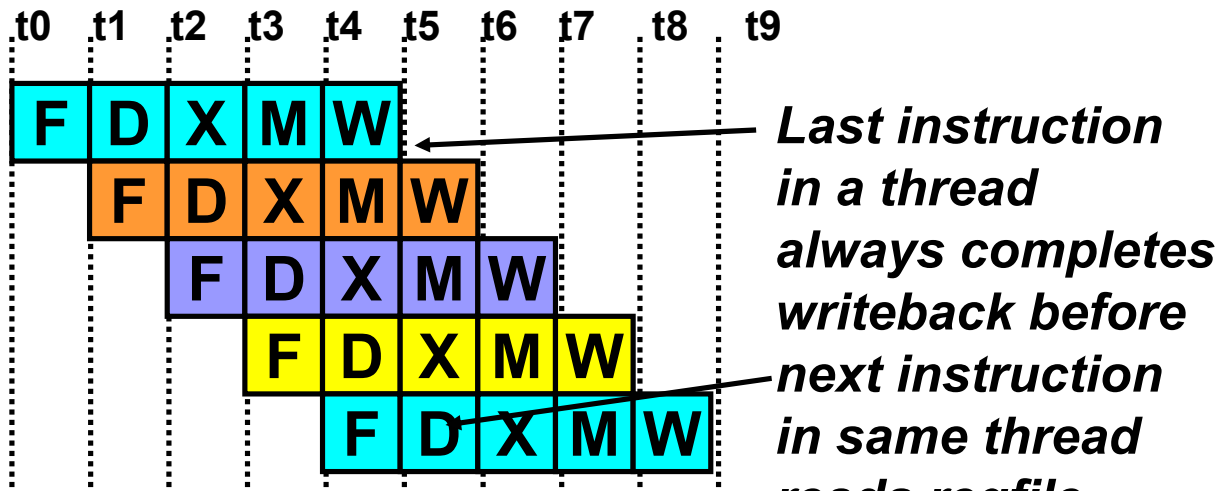
Multithreading

UC Regents Fall 2013 © UCB

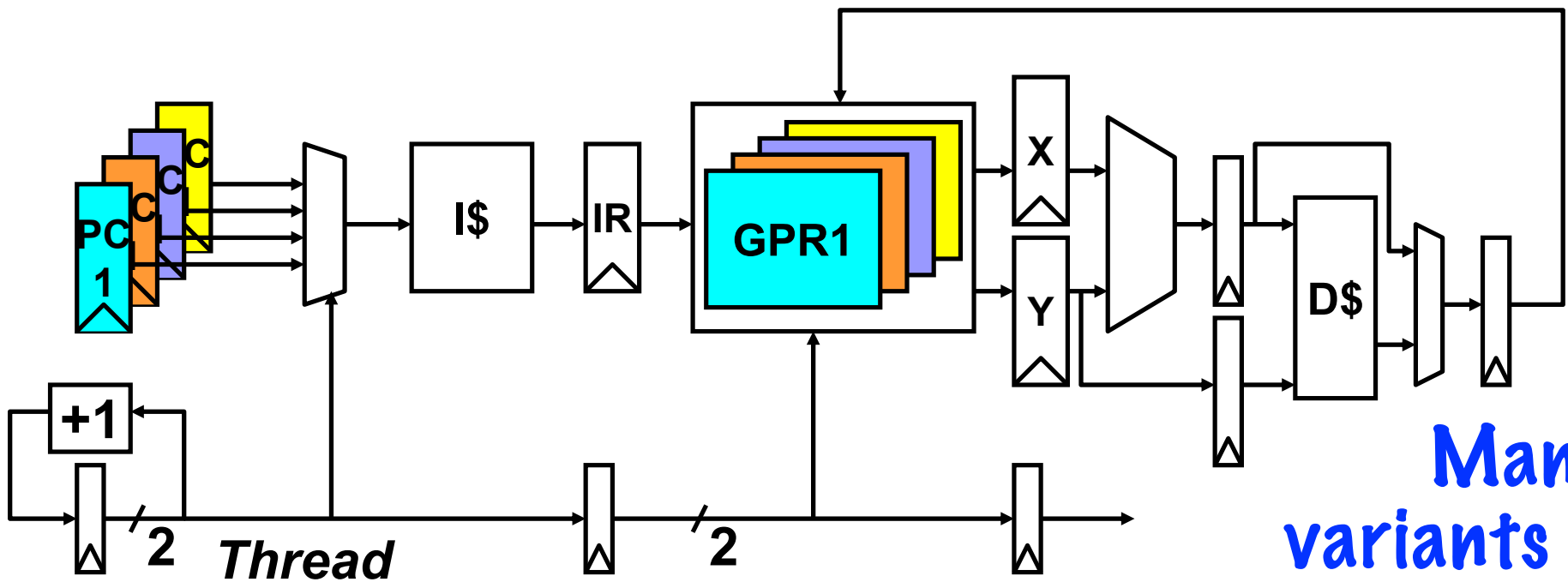
Multithreading of Static Pipelines

Interleave 4 threads, T1-T4, on non-bypassed 5-stage pipe

T1: LW r1, 0(r2)
T2: ADD r7, r1, r4
T3: XORI r5, r4, #12
T4: SW 0(r7), r5
T1: LW r5, 12(r1)



4 CPUs, each run at 1/4 clock



Many variants ...

UC Regents Fall 2013/2014 © UCB

At the logic level …

Synchronous logic we want to "multithread". Critical path is 5.

2X multi-threading: double each register.

Modern synthesis will retime this as shown: critical path is now 2.

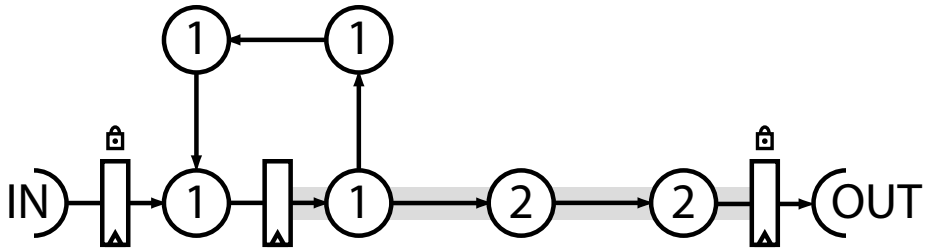


Figure 1: A small graph before retiming. The nodes represent logic delays, with the inputs and outputs passing through mandatory, fixed registers. The critical path is 5.

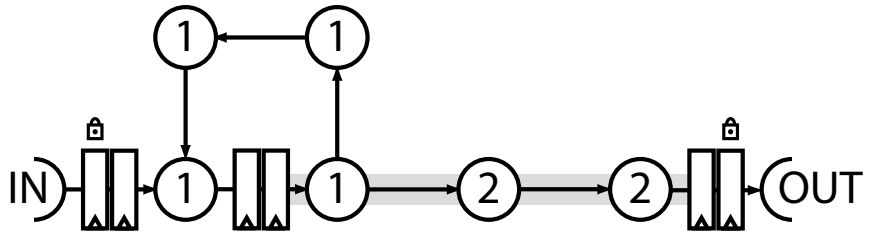


Figure 3: The example in Figure 2 2-slowed. This design now operates on 2 independent data streams.

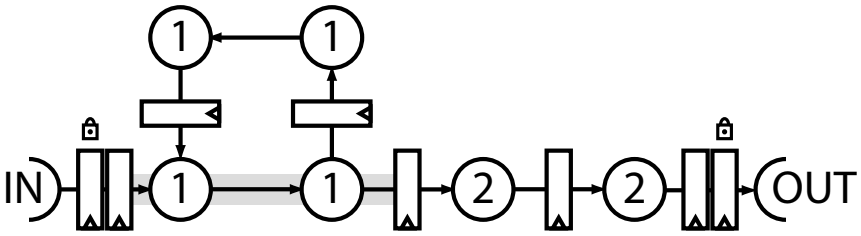


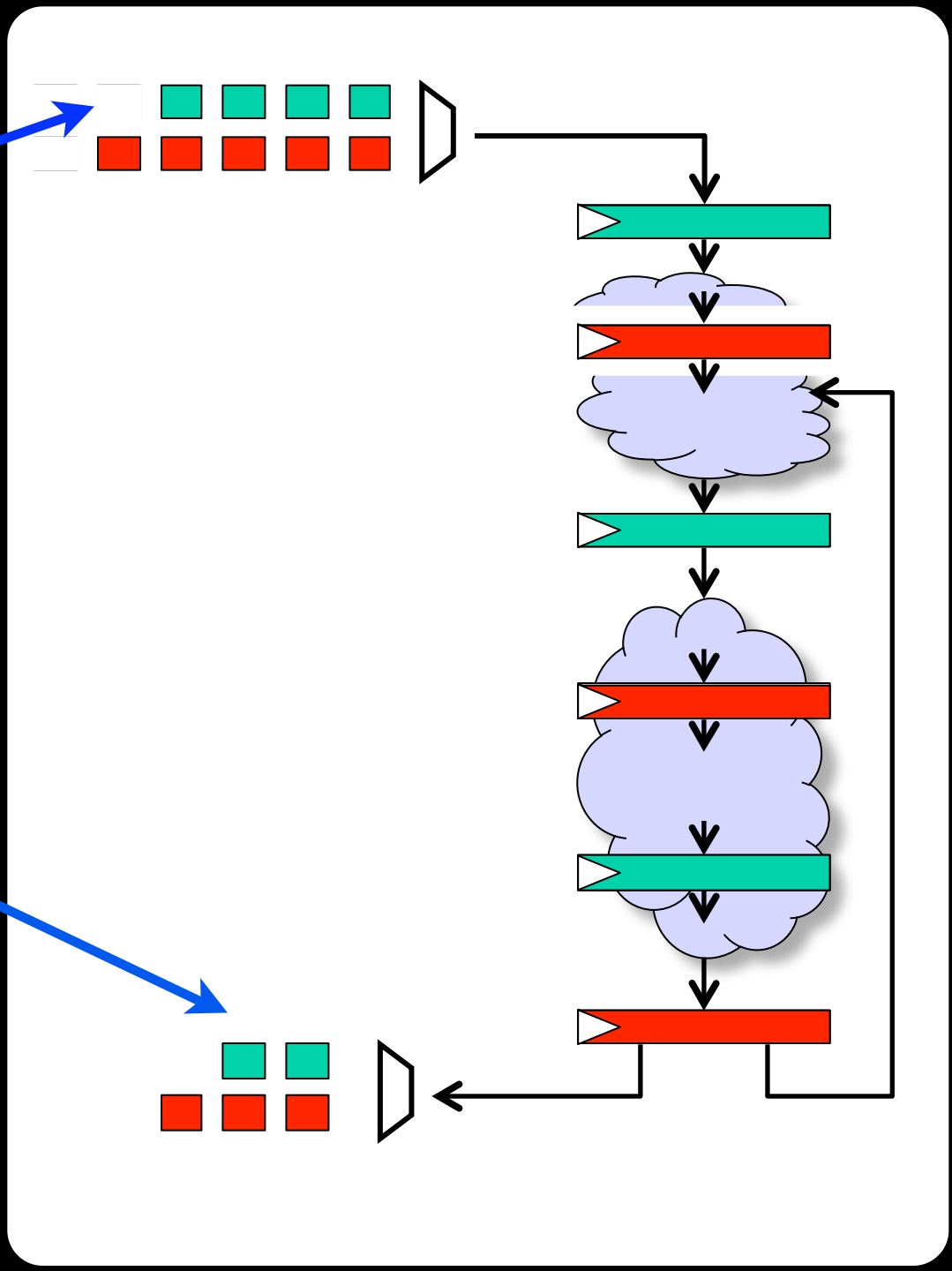
Figure 4: The example in Figure 3 after retiming. The combination of C-slowing and retiming reduced the critical path from 5 to 2.

Nicholas Weaver* UC Berkeley Berkeley, CA
Yury Markovskiy UC Berkeley Berkeley, CA
Yatish Patel UC Berkeley Berkeley, CA
John Wawrzynek UC Berkeley Berkeley, CA

Good fit for GALS

Two input queues (red and green). The mux control logic implements turn-taking.

Outputs placed into two output queues.



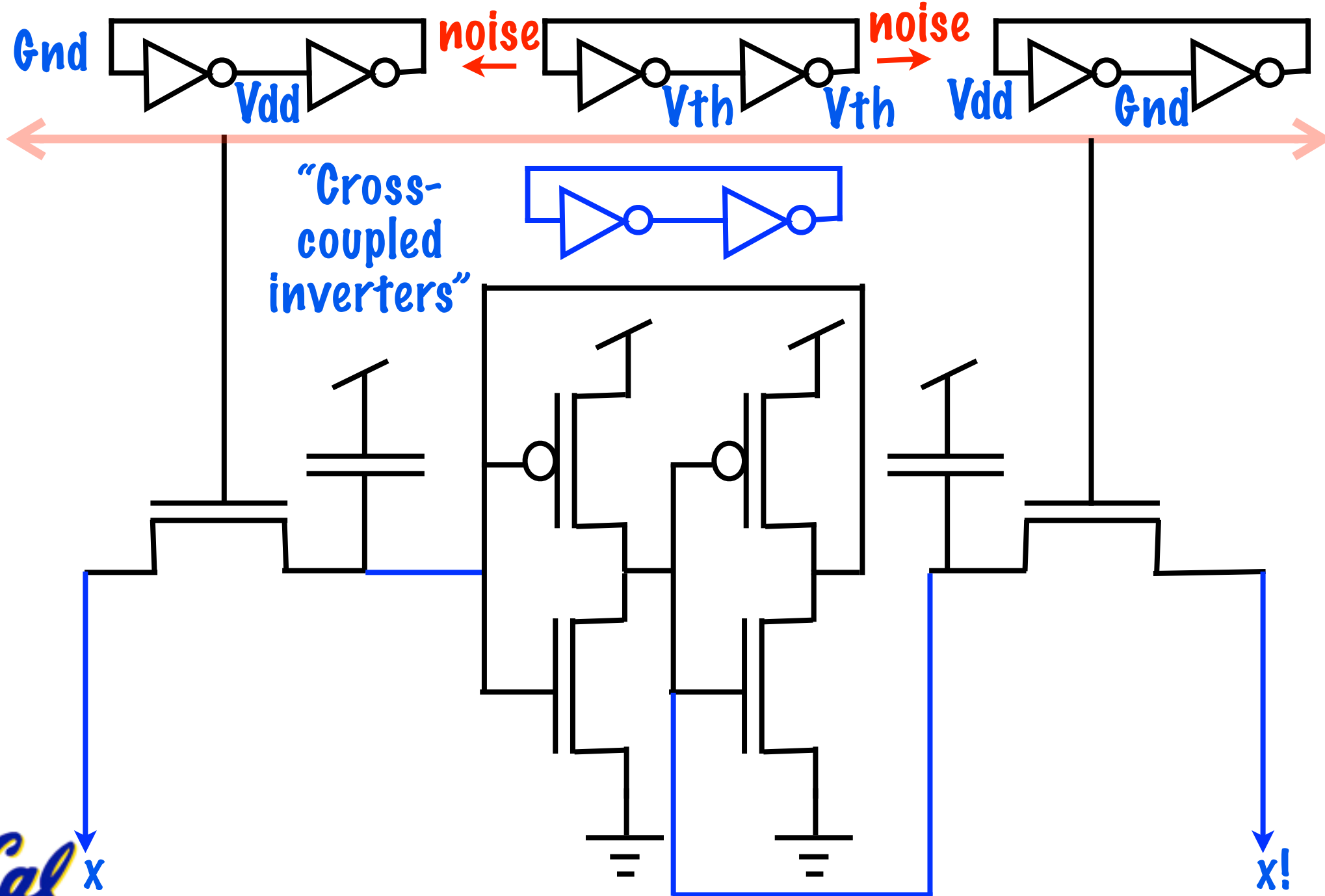
Electrical Details

UC Regents Fall 2013/1014 © UCB

Flip Flops Revisited

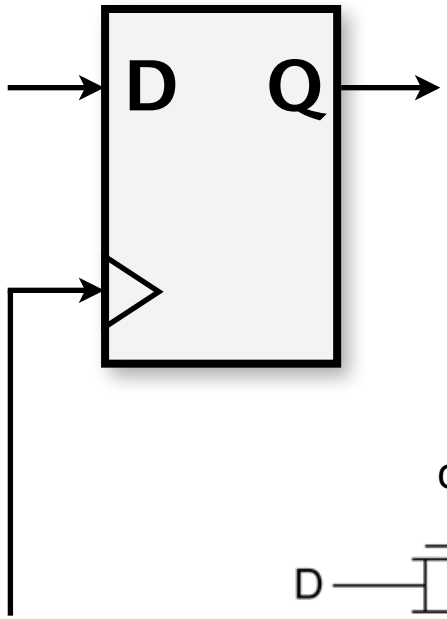
UC Regents Fall 2013 © UCB

Recall: Static RAM cell (6 Transistors)

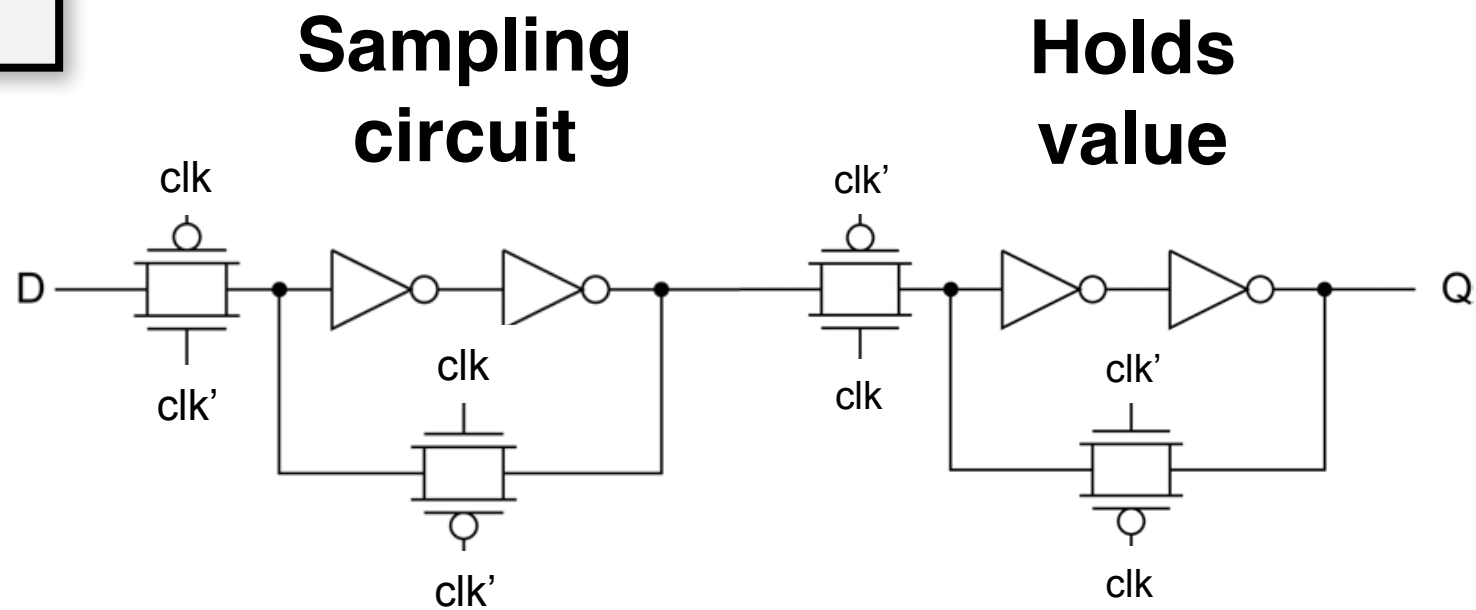


UC Regents Fall 2013/2014 © UCB

Recall: Positive edge-triggered flip-flop



A flip-flop "samples" right before the edge, and then "holds" value.



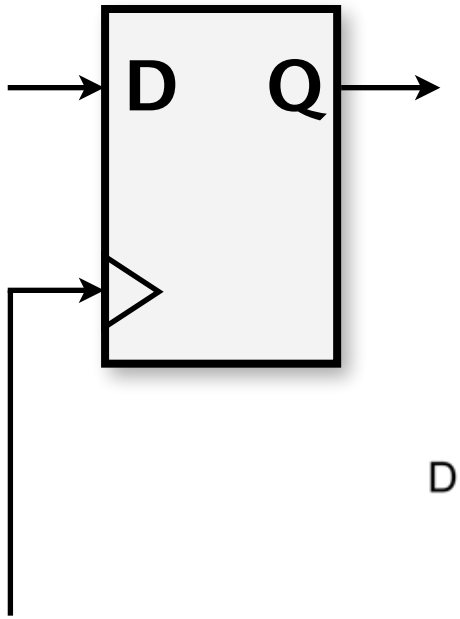
16 Transistors: Makes an SRAM look compact!

What do we get for the 10 extra transistors?

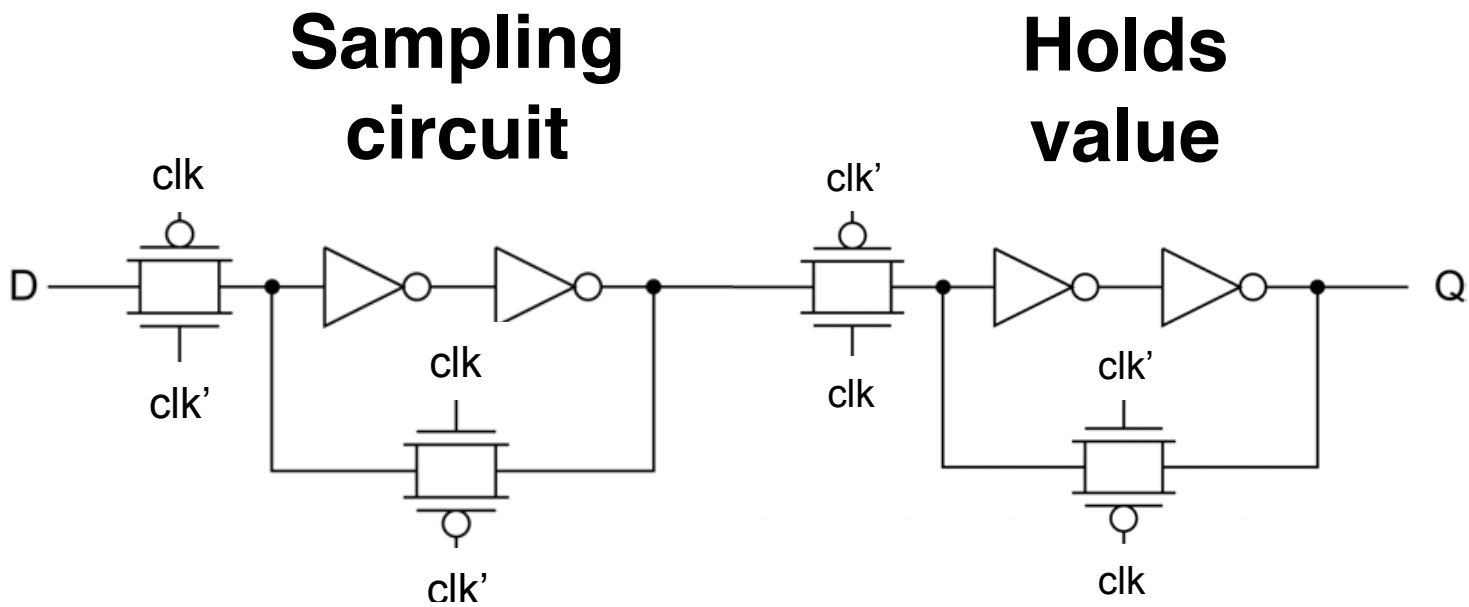
Clocked logic semantics.

UC Regents Fall 2013/2014 © UCB

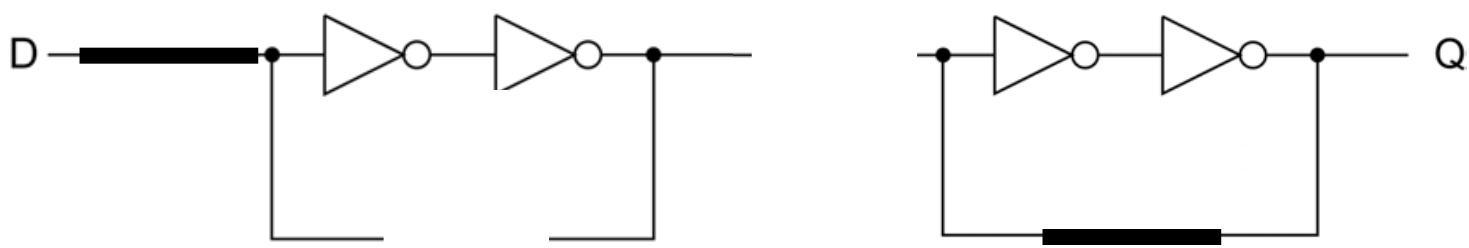
Sensing: When clock is low



A flip-flop "samples" right before the edge, and then "holds" value.



clk = 0
clk' = 1

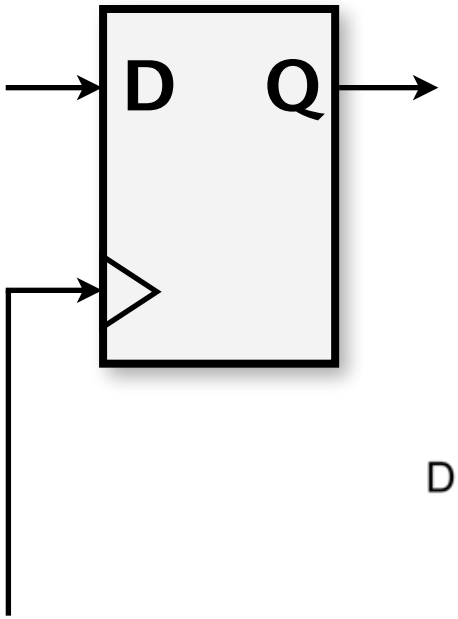


Will capture new value on posedge.

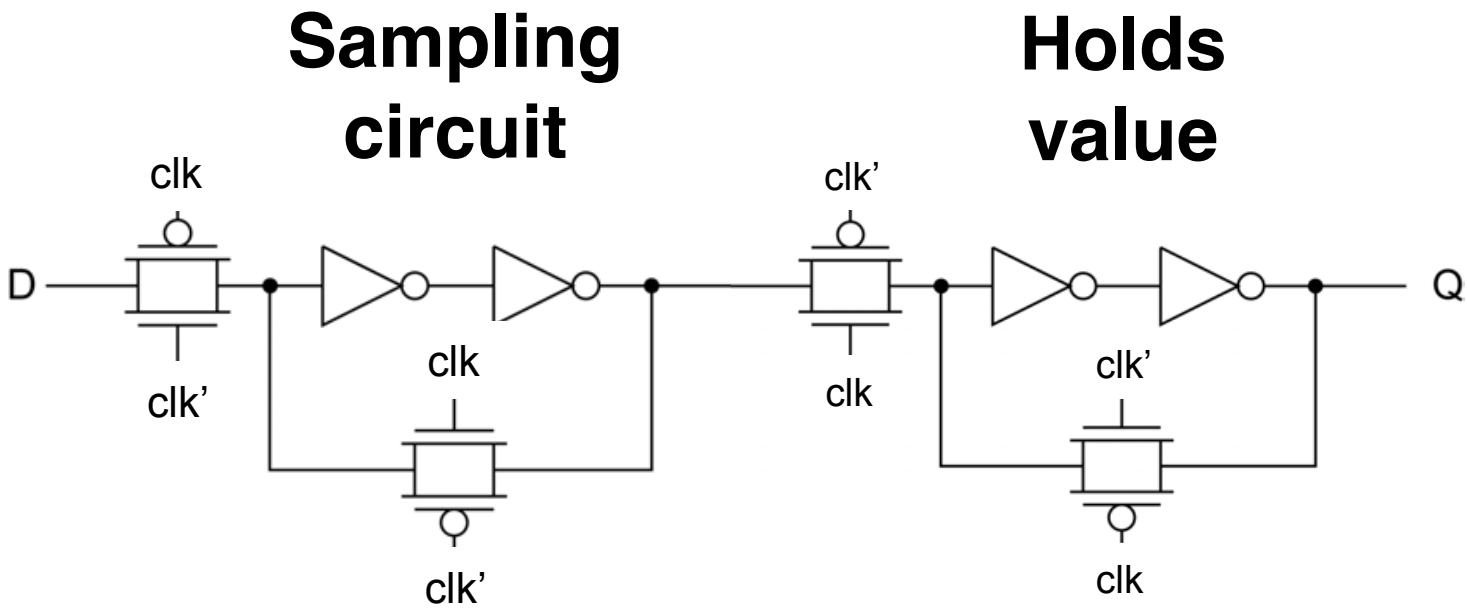
Outputs last value captured.

UC Regents Fall 2013/2014 © UCB

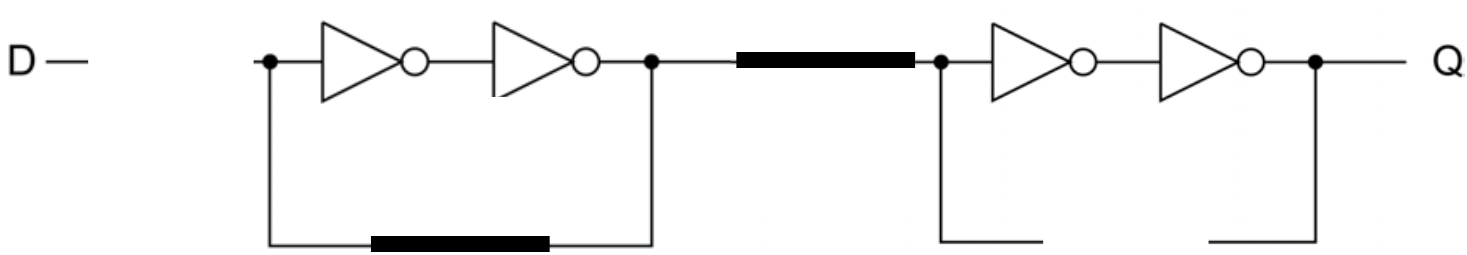
Capture: When clock goes high



A flip-flop "samples" right before the edge, and then "holds" value.



clk = 1
clk' = 0

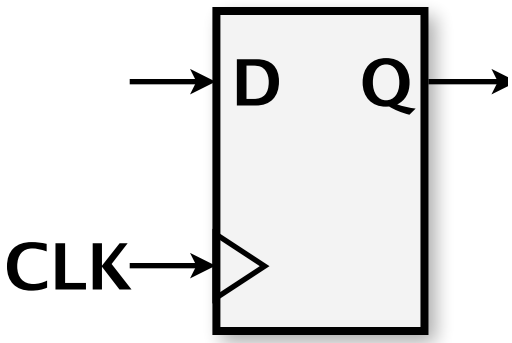


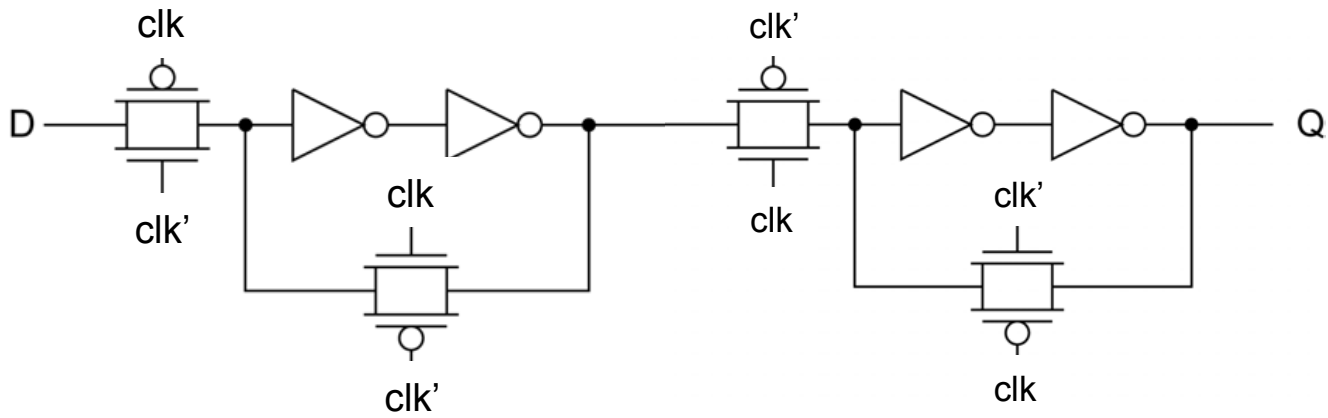
Remembers value just captured.

Outputs value just captured.

UC Regents Fall 2013/2014 © UCB

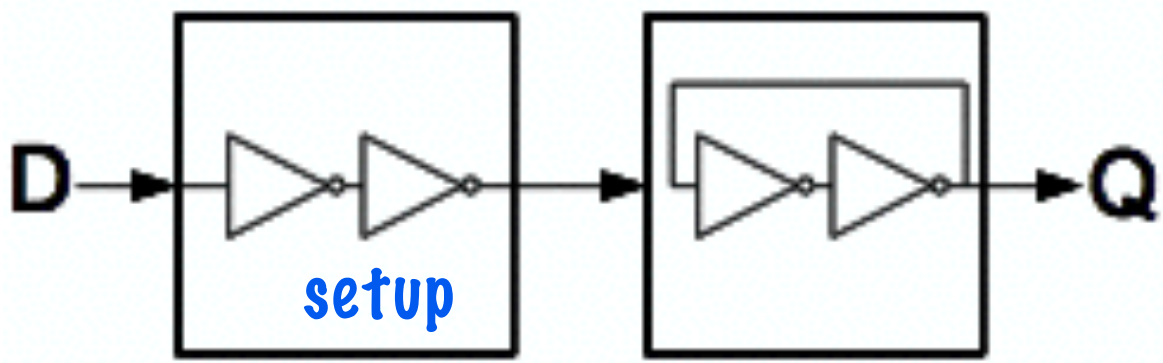
Flip Flop delays: clk-to-Q ? setup ? hold ?





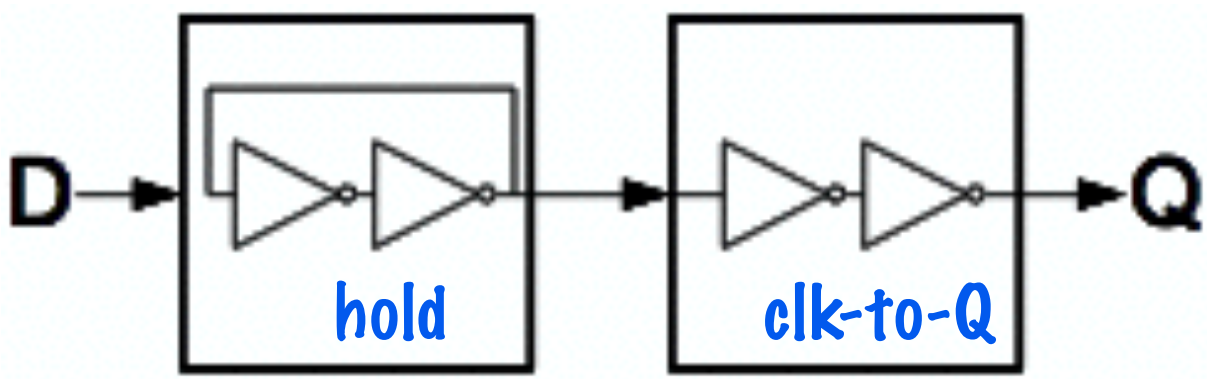
CLK == 0

Sense D, but Q outputs old value.



CLK 0->1

Capture D, pass value to Q



UC Regents Fall 2013/2014 © UCB

On Tuesday … Power and Energy