EECS 151/251A Spring 2023 Digital Design and Integrated Circuits

Instructor:
John Wawrzynek

Lecture 11: Timing Part 1

Announcements

❏ Midterm exam coming up in 2 weeks

What do ASIC/FPGA Designers need to know about physics?

▸ Physics effect:

Area $\Rightarrow$ cost
Delay $\Rightarrow$ performance
Energy $\Rightarrow$ performance & cost

- Ideally, zero delay, area, and energy. However, the physical devices occupy area, take time, and consume energy.

- CMOS process lets us build transistors, wires, connections, and we get capacitors, inductors, and resistors whether or not we want them.
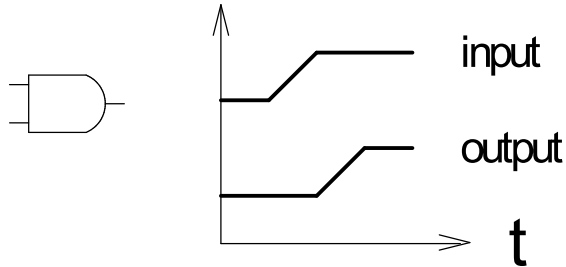
Performance, Cost, Power

clock ⎍⎍   input

input → CL → reg → CL → reg → output ...
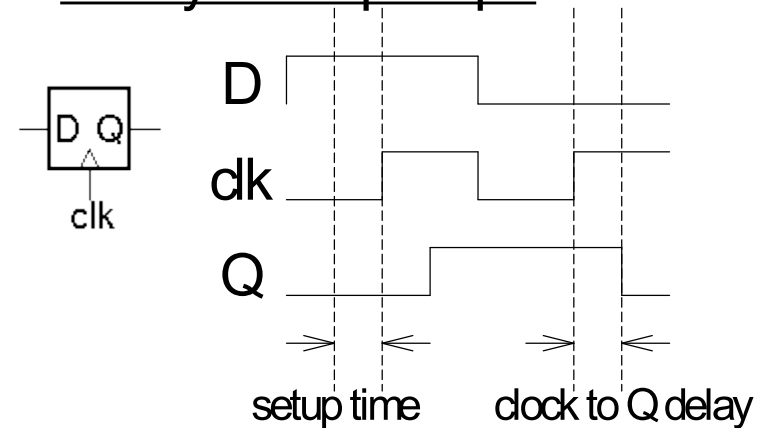
option feedback

output
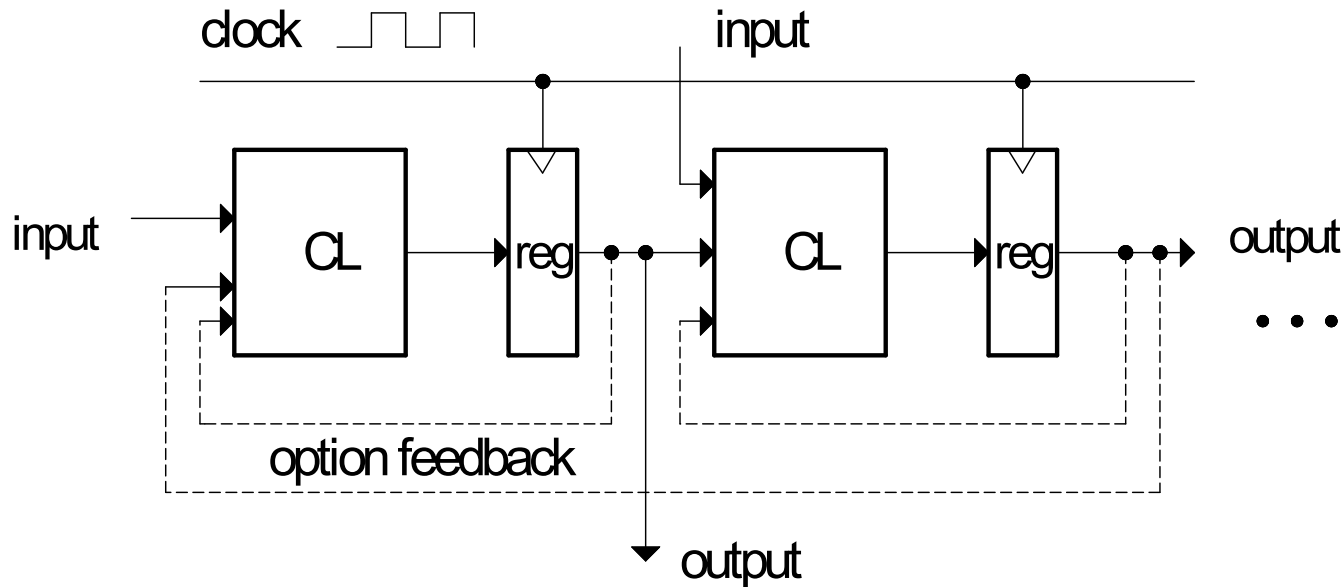
- How do we measure performance?

  operations/sec? cycles/sec?

- Performance is directly proportional to clock frequency. Although it may not be the entire story:

  Ex: CPU performance

  = # instructions X CPI X clock period

Limitations on Clock Rate

1   Logic Gate Delay

input

output

t

2   Delays in flip-flops

D Q
clk

D

clk

Q

setup time   clock to Q delay
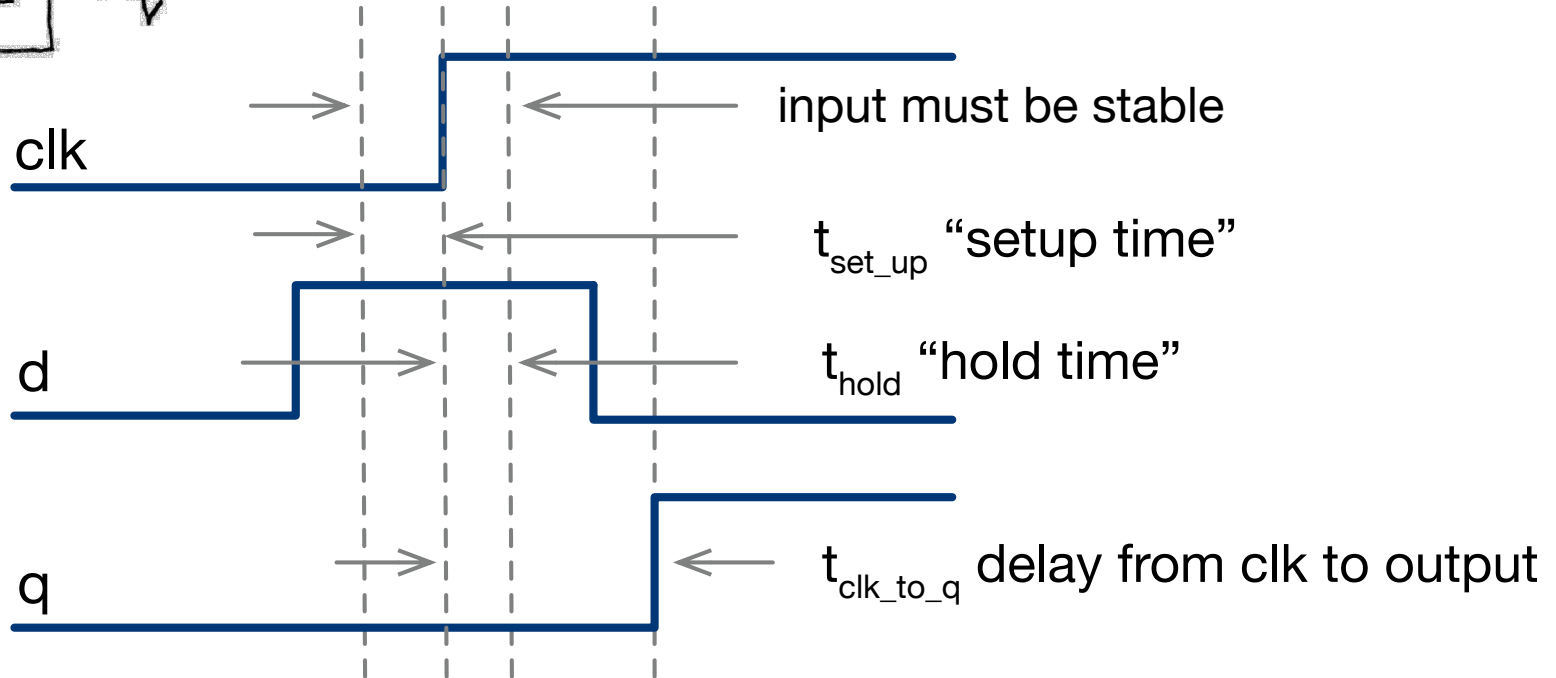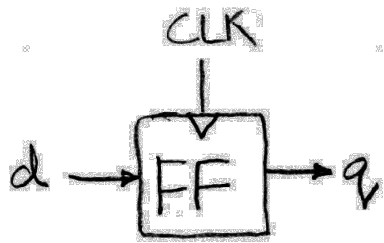
3   Interconnect Delay: wires

v1   v2   v3   v4
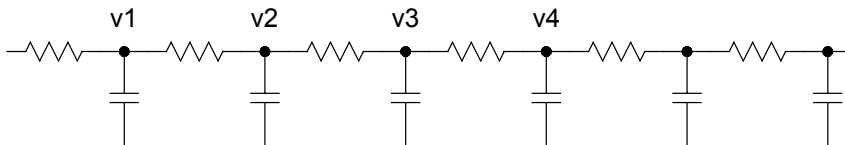
1, 2, & 3 all contribute to limiting the clock period.

- What must happen in one clock cycle for correct operation?
  – All signals connected to FF (or memory) inputs must be ready and "setup" before rising edge of clock.
  – For now we assume perfect clock distribution (all flip-flops see the clock at the same time).

Register Timing Details

clk

input must be stable

t_{set_up} "setup time"

d

t_{hold} "hold time"
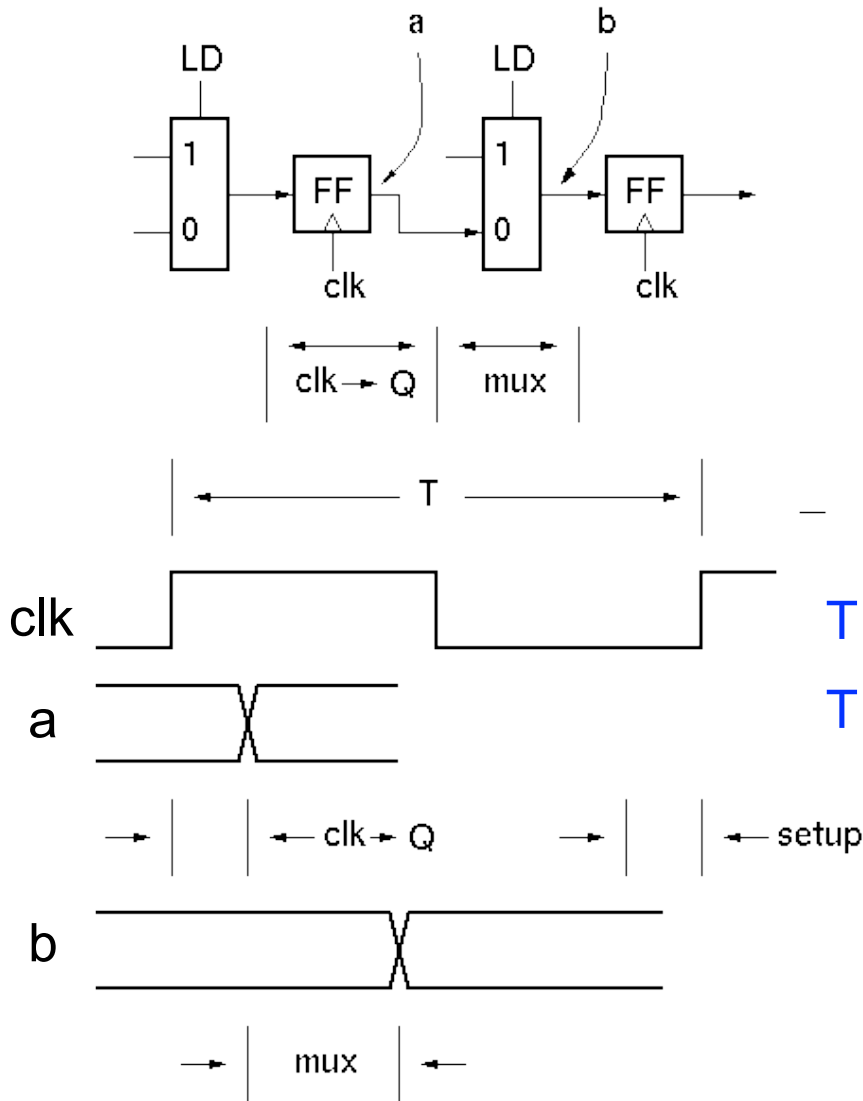
q

$t_{clk\_to\_q}$ delay from clk to output

❑ Three important times associated with flip-flops:

- Setup time - *How long d must be stable before the rising edge of CLK*
- Hold time - *How long D must be stable after the rising edge of CLK*
- Clock-to-q delay – *Propagation delay after rising edge of the CLK*
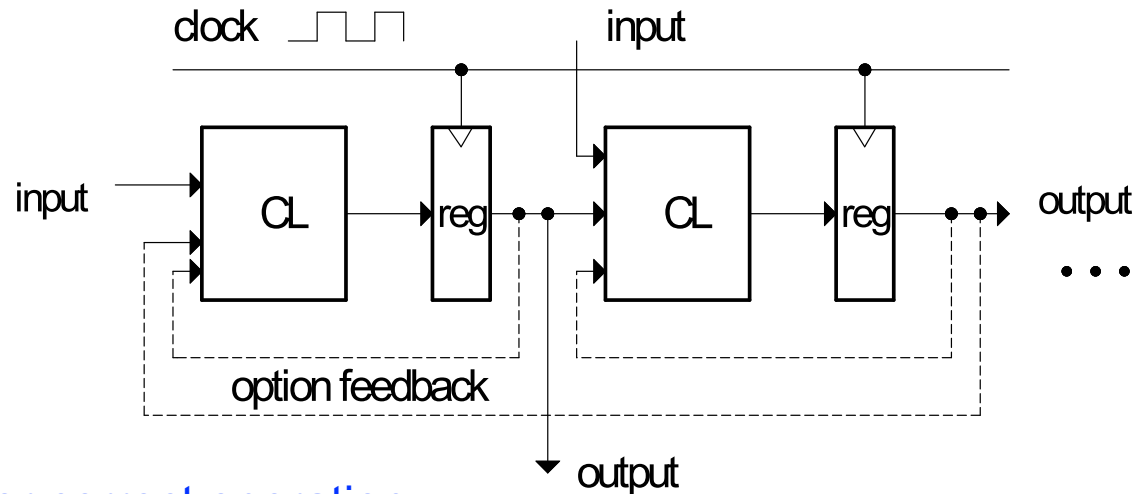
Example: Timing Analysis

Parallel to serial converter circuit

$T \geq time(clk \rightarrow Q) + time(mux) + time(setup)$

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

In General ...

For correct operation:

$$T \geq \tau_{clk \rightarrow Q} + \tau_{CL} + \tau_{setup}$$ for all paths.

- ## How do we enumerate *all* paths?
  - Any circuit input or register output to any register input or circuit output?
- ## Note:
  - "setup time" for outputs is a function of what it connects to.
  - "clk-to-q" for circuit inputs depends on from where it comes.

"Gate Delay"
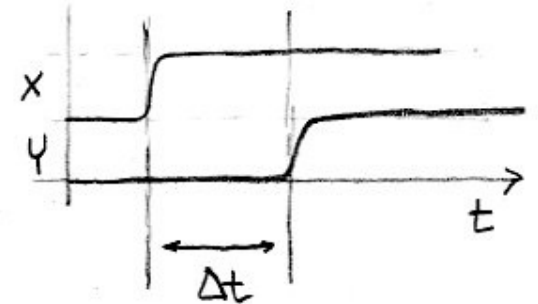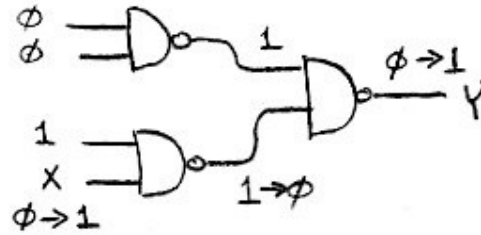
❑ Modern CMOS gate delays on the order of a few picoseconds. (However, highly dependent on gate design and context.)

❑ Often expressed as FO4 delays (fan-out of 4) - as a process dependent delay metric:

- the delay of an **inverter**, driven by an inverter 4x smaller than itself, and driving an inverter 4x larger than itself.

- Less than 10ps for a 32nm process. For a 7nm process FO4 is around 2.5ps.
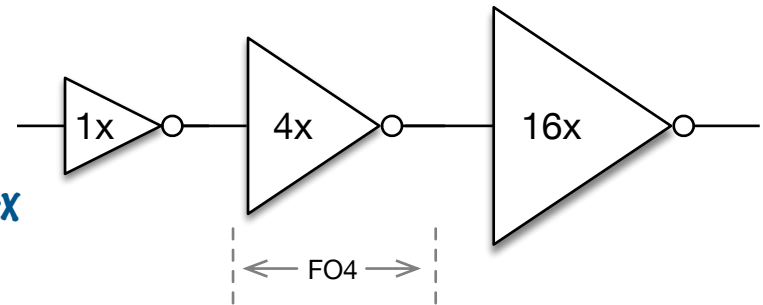
Process Dependent FO4 Delay

Scaling equations for the accurate prediction of CMOS device performance from 180 nm to 7 nm

Aaron Stillmaker[a,b,*], Bevan Baas[a]

[a] Department of Electrical and Computer Engineering, University of California, Davis, One Shields Ave., Davis, CA 95616, USA
[b] Department of Electrical and Computer Engineering, California State University, Fresno, 2320 E. San Ramon Ave., Fresno, CA 93740, USA

Characteristics of different technology nodes [23]. The modeled measurements are for a single inverter in an FO4 chain. The energy value is the average energy required for a single inverter transition from low to high, or high to low.

Production Year	Technology Node (nm)	Technology Type	V_{DD} (V)	SimulatedPerformance of Inverter		
				Delay (ps)	Energy (fJ)	Power (μW)
1999	180	Bulk	1.8	77.2	27.5	105
2001	130	Bulk	1.2	34.7	5.20	26.1
2004	90	Bulk	1.1	26.5	2.62	13.0
2007	65	Bulk	1.1	19.8	1.72	8.58
2008	45	High-k	1.1	10.9	1.05	5.19
2010	32	High-k	0.97	9.8	0.51	2.47
2012	20	Multi-Gate	0.9	9.66	0.198	1.51
2013	16[a]	Multi-Gate	0.86	6.12	0.179	1.28
2013	14[a]	Multi-Gate	0.86	4.02	0.144	0.995
2015	10	Multi-Gate	0.83	3.24	0.122	0.866
2017	7	Multi-Gate	0.8	2.47	0.111	0.789

[a] The 2013 ITRS report labels a single "16/14" node.

"Path Delay"

$\Delta t \propto \#$ of series gates
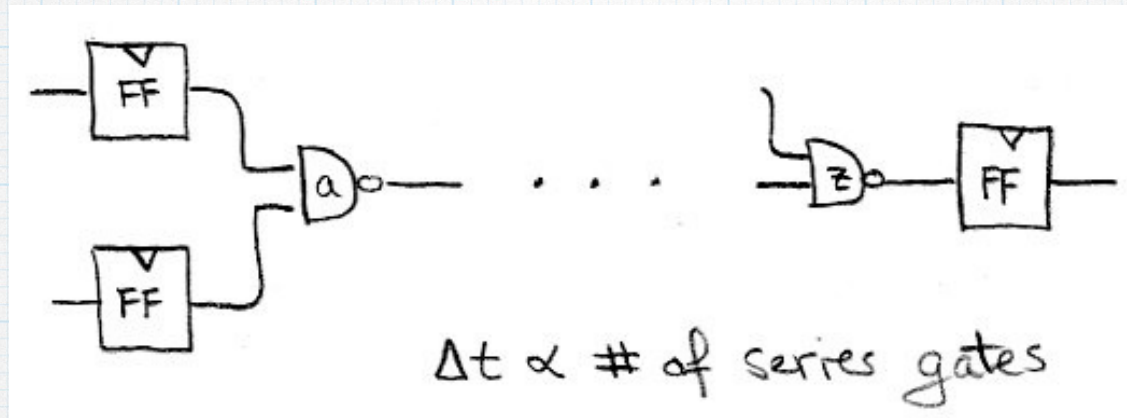
▸ For correct operation:

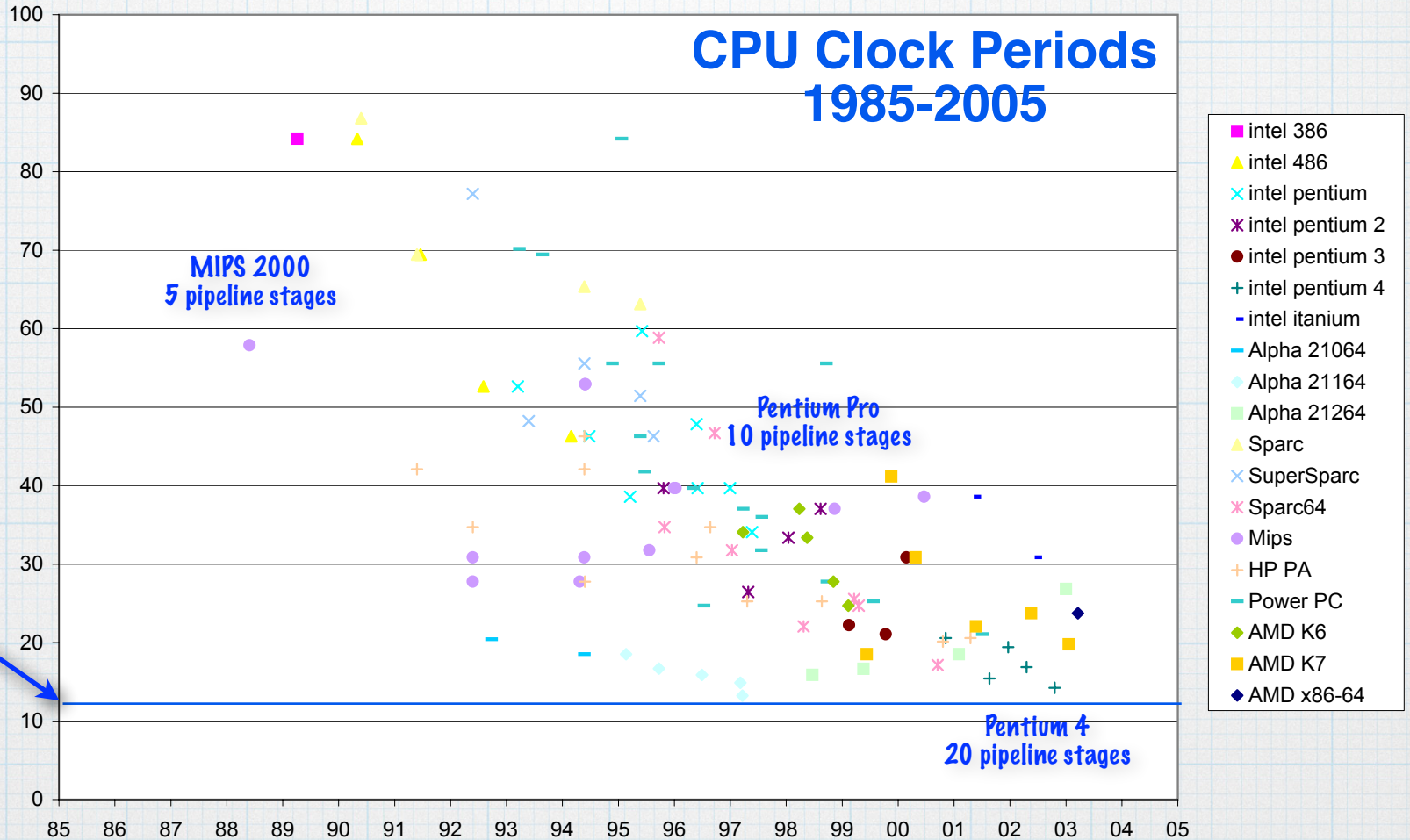Total Delay ≤ clock_period − FF_{setup_time} − $FF_{clk\_to\_q}$

on all paths.

▸ High-speed processors critical paths (worst case paths) have around 20 FO4 delays.

FO4 Delays per clock period

FO4 Delays

CPU Clock Periods
1985-2005

MIPS 2000
5 pipeline stages

Pentium Pro
10 pipeline stages

Historical limit: about 12

Pentium 4
20 pipeline stages

Legend:
- ■ intel 386
- ▲ intel 486
- ✕ intel pentium
- ✳ intel pentium 2
- ● intel pentium 3
- + intel pentium 4
- - intel itanium
- — Alpha 21064
- ◆ Alpha 21164
- ▢ Alpha 21264
- ▲ Sparc
- ✕ SuperSparc
- ✕ Sparc64
- ● Mips
- + HP PA
- — Power PC
- ◆ AMD K6
- ■ AMD K7
- ◆ AMD x86-64

Thanks to Francois Labonte, Stanford

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

F04 Delays Per Cycle for Processor Designs

F04 delay per cycle is roughly proportional to the amount of computation completed per cycle.

"Gate Delay"

- ❑ What determines the actual delay of a logic gate?
- ❑ Transistors are not perfect switches - cannot change terminal voltages instantaneously.
- ❑ Consider the NAND gate:

  - ▪ Current (I) value depends on: process parameters, transistor size
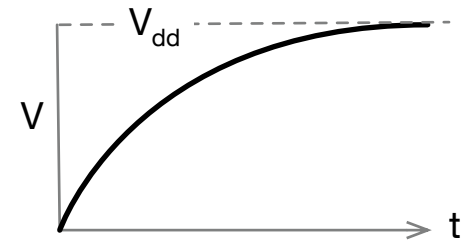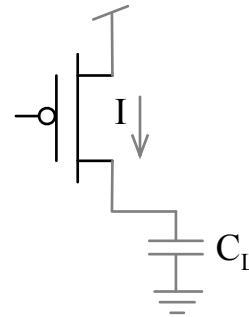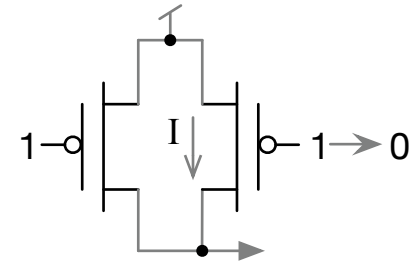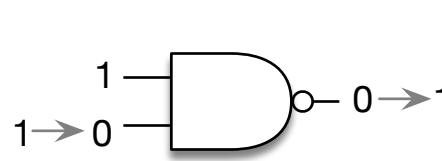
$$\Delta t \propto C_L / I$$

- ▸ C_L models gate output, wire, inputs to next stage (Cap. of Load)
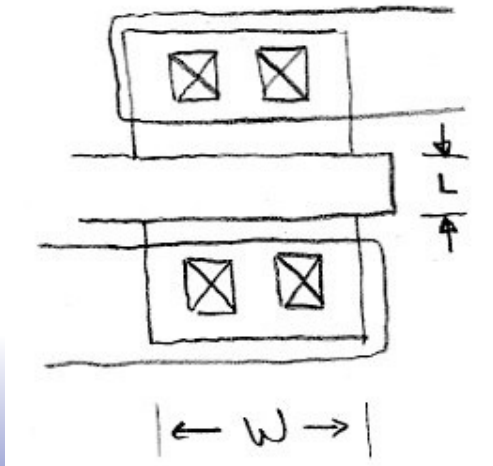- ▸ C "integrates" I creating a voltage change at output

14

More on transistor Current

❑ Transistors actually act like a cross between a resistor and "current source"
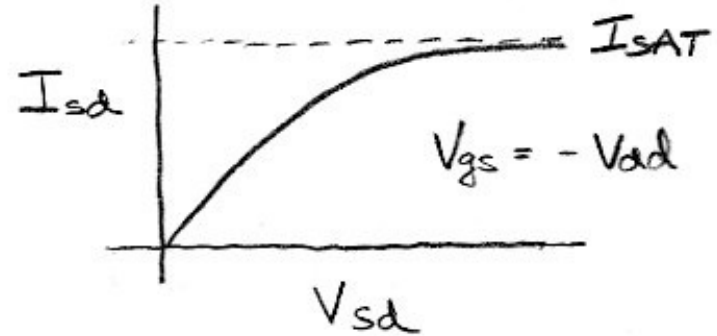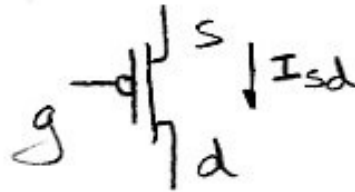
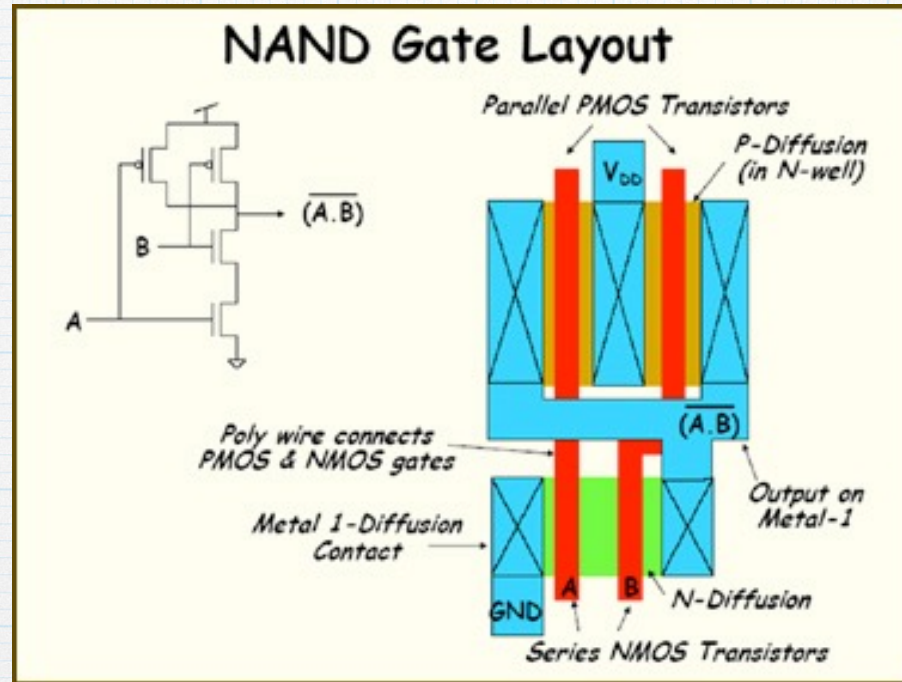▸ I_{SAT} depends on process parameters (higher for nFETs than for pFETs) and transistor size (layout):

$$I_{SAT} \propto W/L$$

FinFets use multiple "fins" to get wider
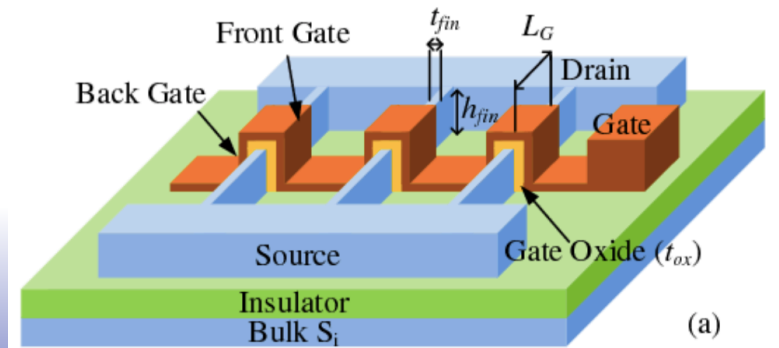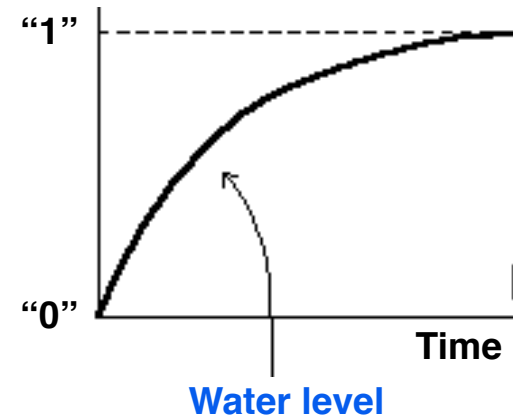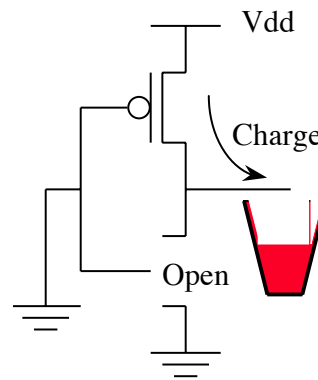
Physical Layout determines FET strength

NAND Gate Layout

- ▸ "Switch-level" abstraction gives a good way to understand the **function** of a circuit.
  - ▸ nFET (g=1 ? short circuit : open)
  - ▸ pFET (g=0 ? short circuit : open)
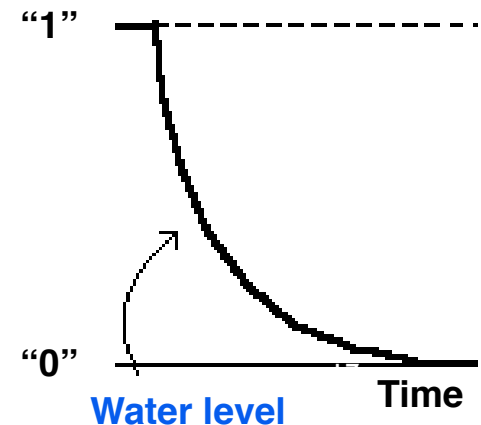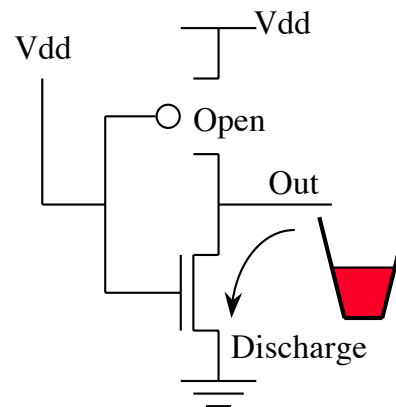- ▸ Understanding delay means going below the switch-level abstraction to transistor physics and layout details.

...tor) transistors
...ctor) transistors

Vdd = 5V

GND = 0v

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

Vdd = 5V

GND = 0v

...fills
...citor
...e.

CS152 / Kubiatowicz
Lec3.31

Vdd

Charge

Open

"1"

"0"

Time

Water level

A "on" n-FET
empties the bucket.

Vdd

Vdd

Open

Out

Discharge

"1"

"0"

Time

Water level

Inverter: Transient Response

With:

resistive approximation for FETs,
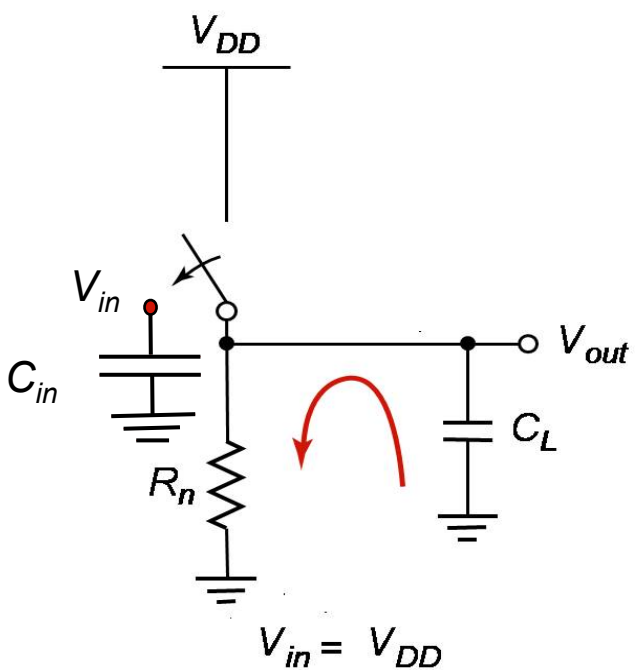high-to-low (HL) $\longrightarrow$

$$V(t) = V_0\, e^{-t/RC}$$

$$t_{1/2} = \ln(2) \times RC$$
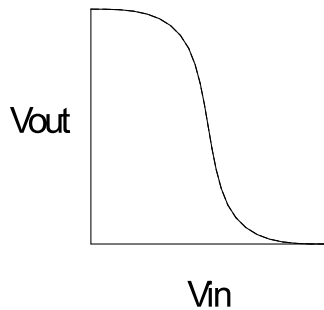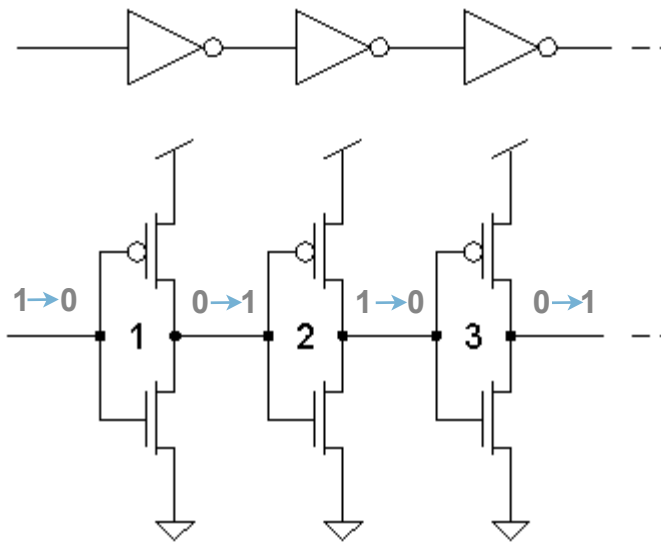
$$t_{pHL} = f(R_{on}C_L)$$
$$= 0.69\, R_n C_L$$
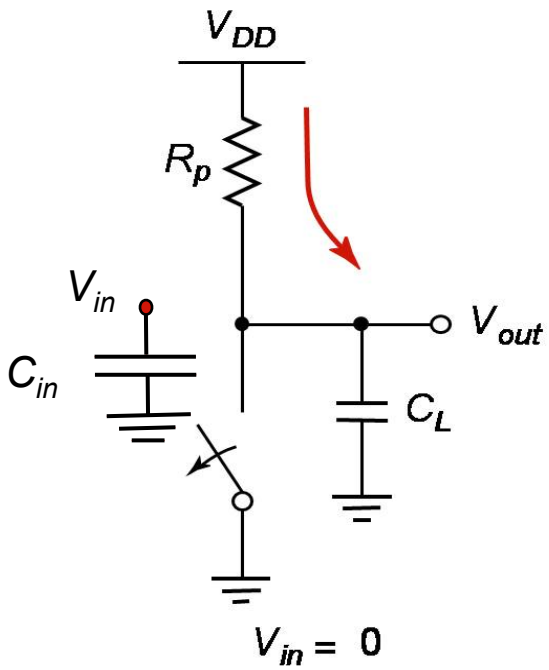
(a) Low-to-high

(b) High-to-low

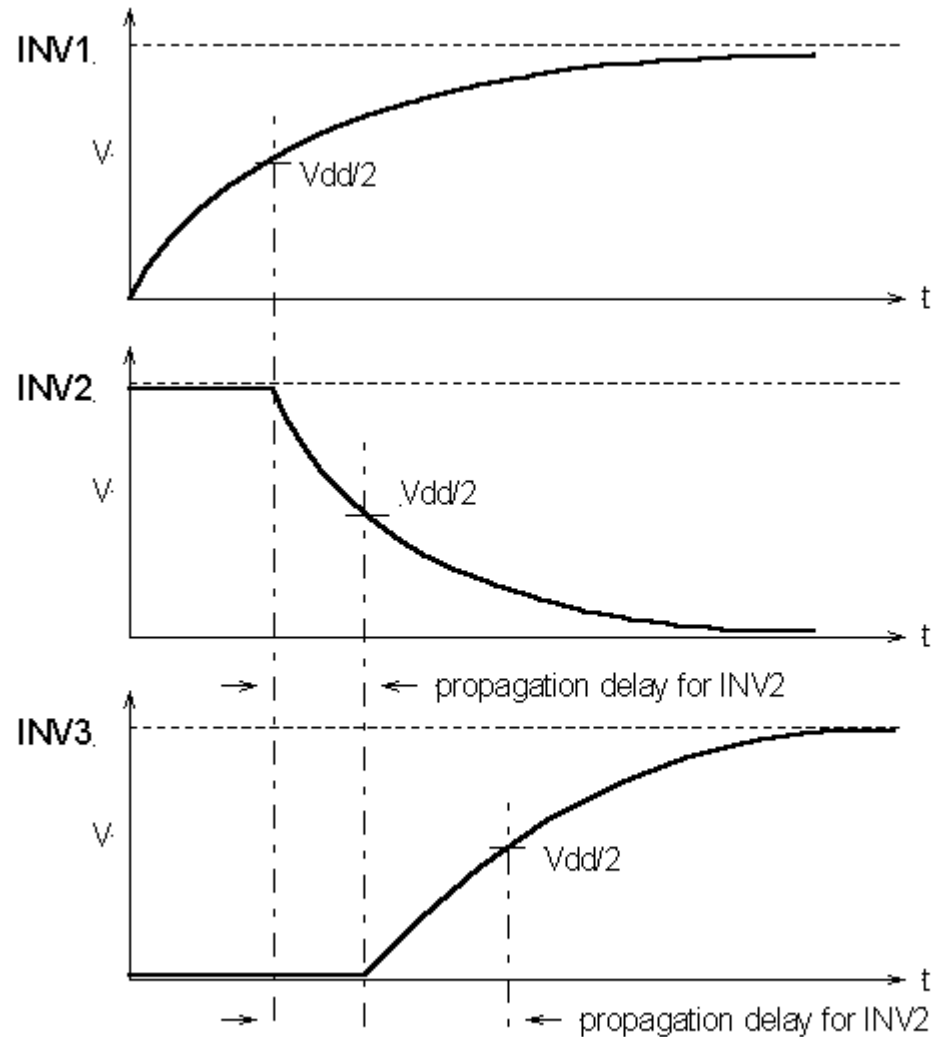# *Turning Rise/Fall Delay into Gate Delay*

- Cascaded gates:

$1 \rightarrow 0$ $0 \rightarrow 1$ $1 \rightarrow 0$ $0 \rightarrow 1$

Vout

Vin

"transfer curve" for inverter.

INV1

V   Vdd/2

INV2

V   Vdd/2

→| |← propagation delay for INV2

INV3

V   Vdd/2

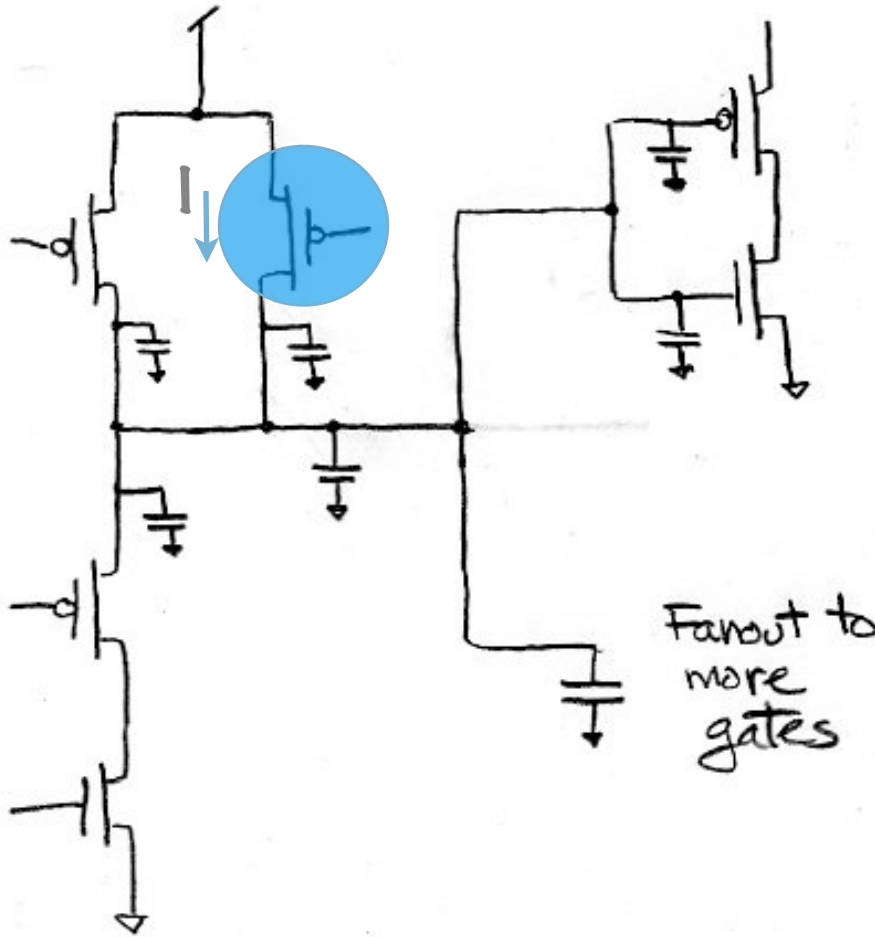→| | propagation delay for INV2 & INV3 in series

In general:

prop. delay = sum of individual prop. delays of gates in series.

19

More on gate delay

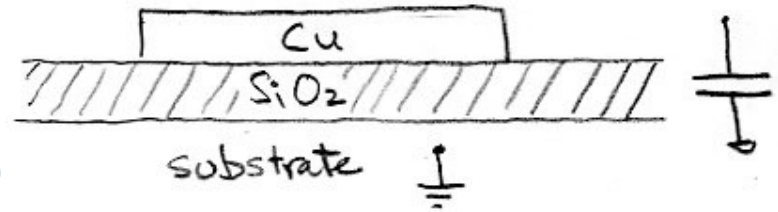❑ Everything that connects to the output of a logic gate (or transistor) contributes capacitance:

▸ **Transistor drains**

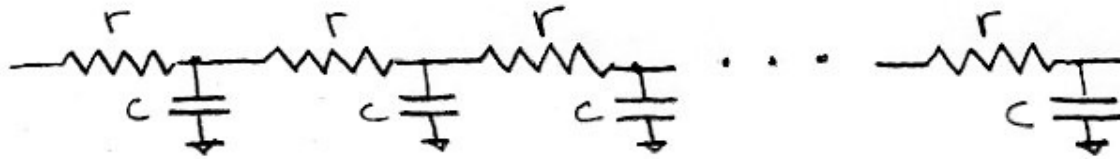▸ **Interconnection (wires/contacts/ vias)**

▸ **Transistor Gates**

Wires

❑ As parallel plate capacitors:
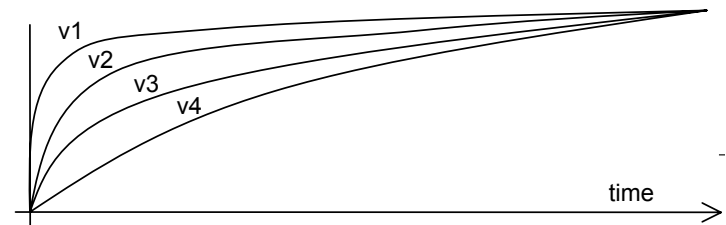
$$C \propto Area = width * length$$

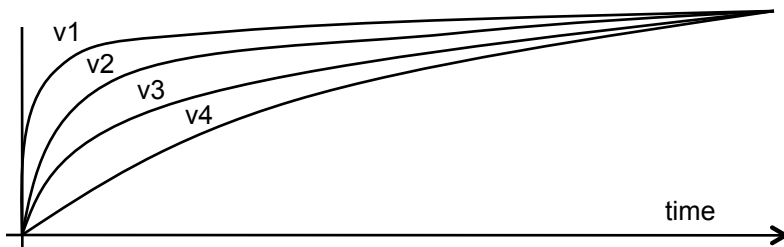▸ **Wires have some finite resistance, so have distributed R and C:**

with r = res/length, c = cap/length, $\Delta t \propto rcL^2 \cong rc + 2rc + 3rc + ...$
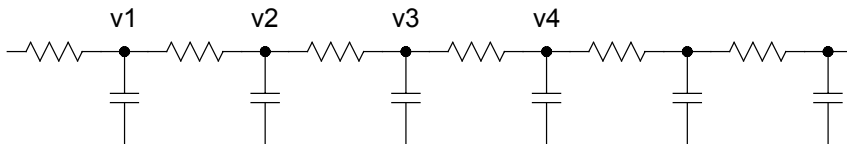
Wire Delay

- Wires posses distributed resistance and capacitance

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

v1   v2   v3   v4

time
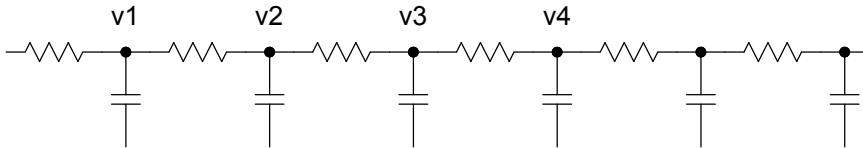
- For **short wires** on ICs, resistance is insignificant (relative to effective R of transistors), but C is important.
  - Typically around half of C of gate load is in the wires.
- For **long wires** on ICs:
  - busses, clock lines, global control signal, etc.
  - Resistance is significant, therefore distributed RC effect dominates.
  - signals are typically "rebuffered" to reduce delay
- For **long wires** on ICs with high currents:
  - *inductance* is also important

Wire Rebuffering

- For **long wires** on ICs:
  - busses, clock lines, global control signal, etc.
  - Resistance is significant, therefore rcL^2 effect dominates.
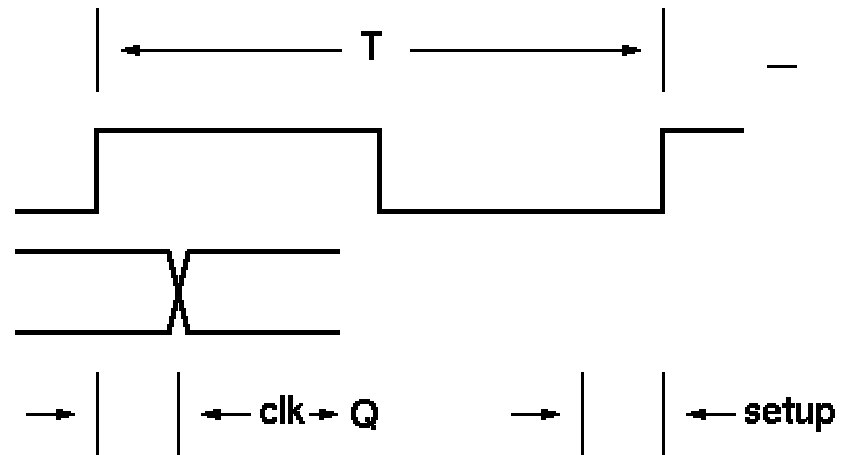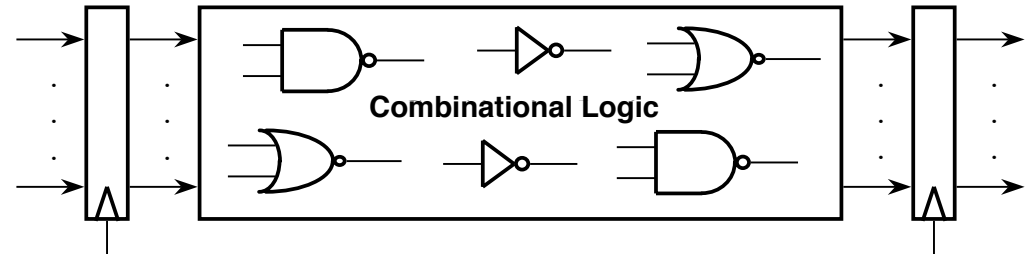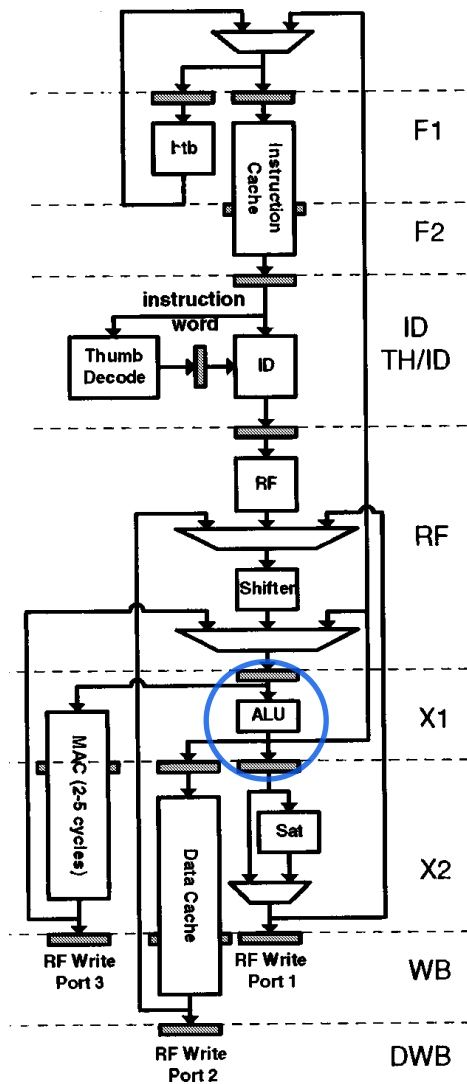 - signals are typically "rebuffered" to reduce delay:
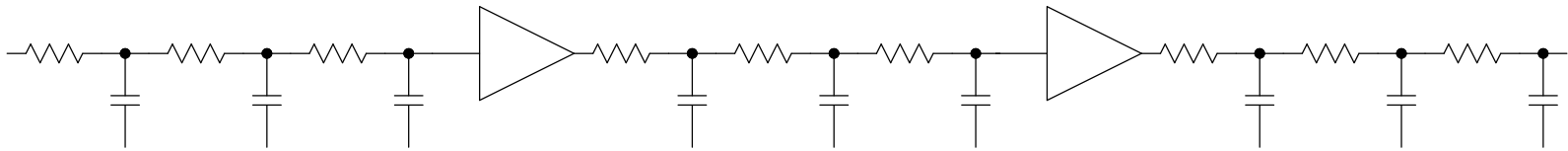
unbuffered wire $\Delta t \propto L^2$

wire buffered into N sections $\Delta t \propto N * (L/N)^2 + (N-1) * t_{buffer}$

Assuming t_{buffer} is small, $\Delta t \propto L^2/N$
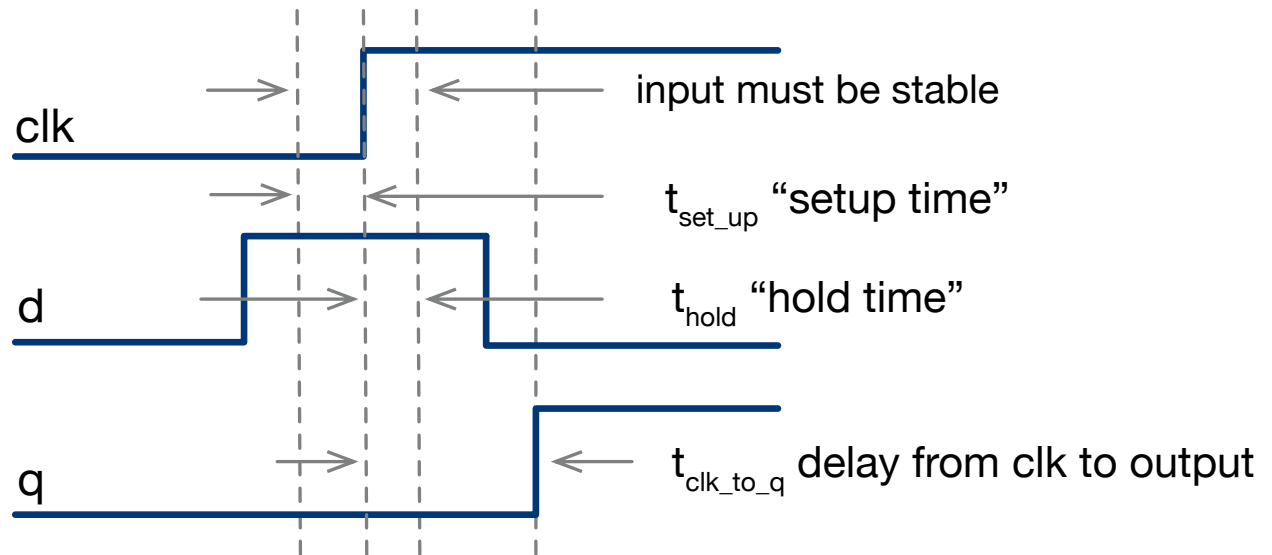
Speedup: $\propto N$

Flip-Flop delays eat into "time budget"

Combinational Logic

ALU "time budget"

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

Recall: Positive edge-triggered flip-flop

Sampling circuit

Holds value

D ⊳ clk / clk' ▷○ ▷○ ... clk' / clk ▷○ ▷○ Q

input must be stable

clk

t_{set_up} "setup time"

d

t_{hold} "hold time"

q

$t_{clk\_to\_q}$ delay from clk to output
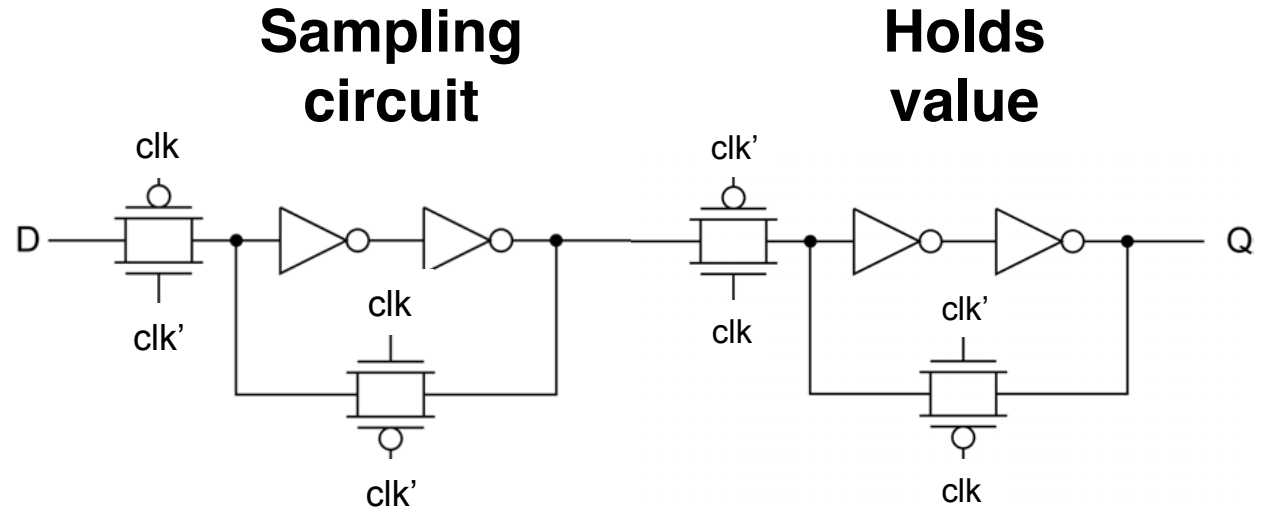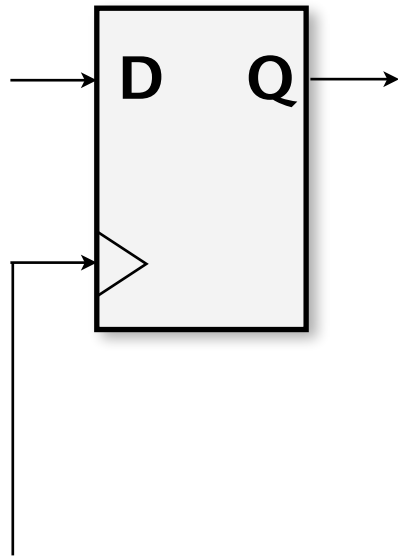
Sensing: When clock is low
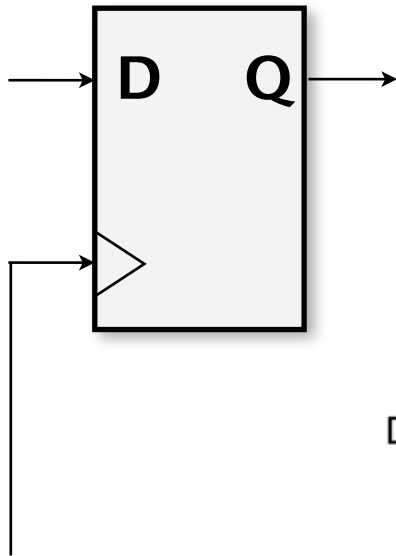
A flip-flop "samples" right before the edge, and then "holds" value.

Sampling circuit

Holds value

D ... Q
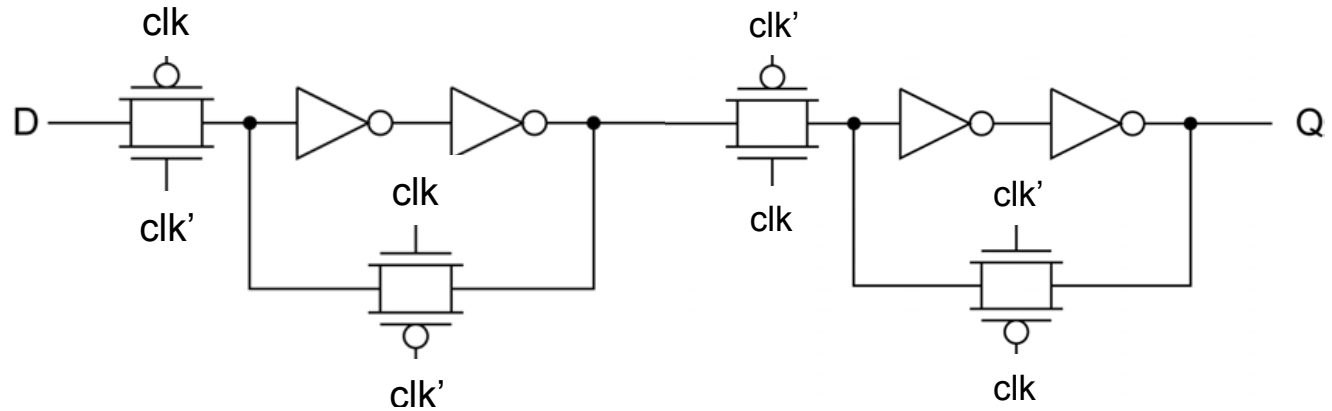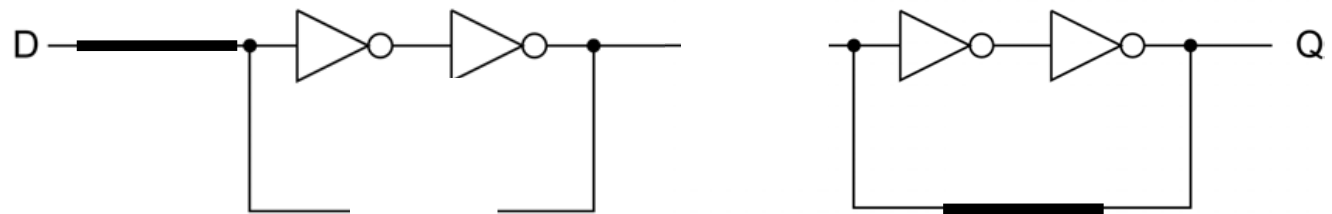
clk' = 0
clk' = 1

Will capture new value on posedge.

Outputs last value captured.

Capture: When clock goes high

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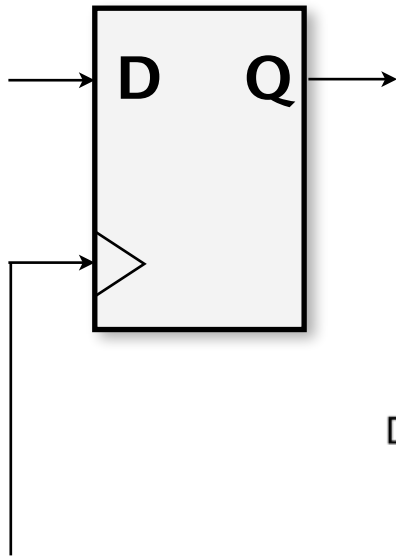
Sampling circuit

Holds value
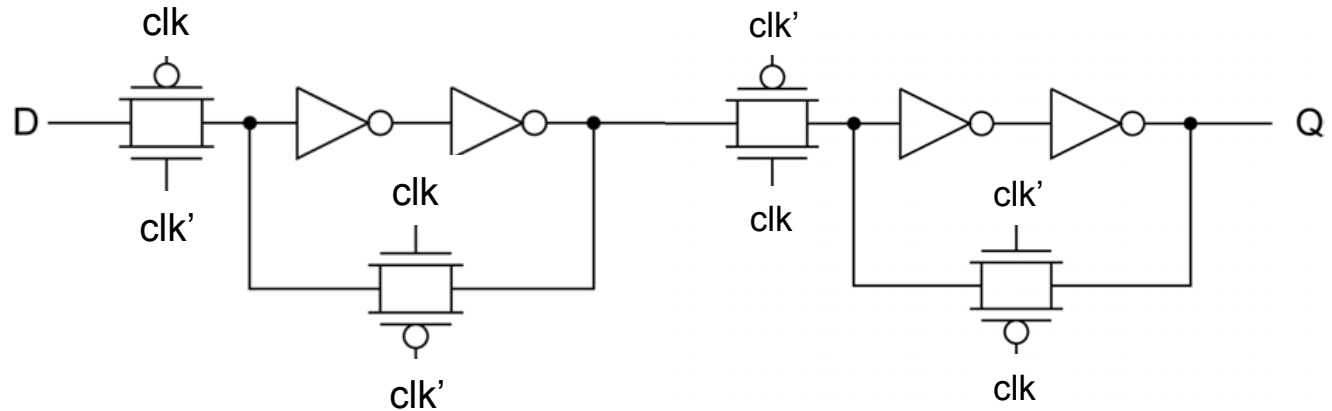
clk = 1
clk' = 0

Remembers value just captured.

Outputs value just captured.

Flip Flop delays: clk-to-Q ?   setup ?
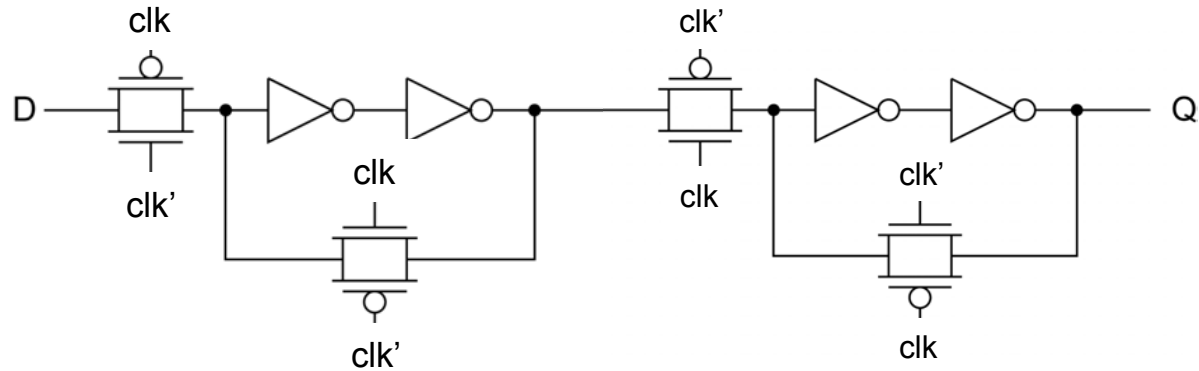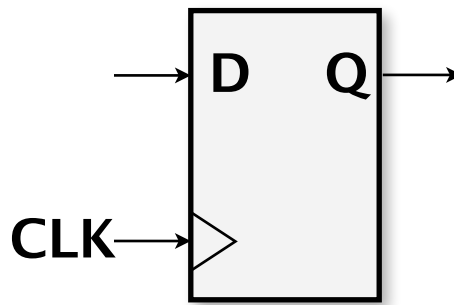
CLK == 0

Sense D, but Q outputs old value.

setup
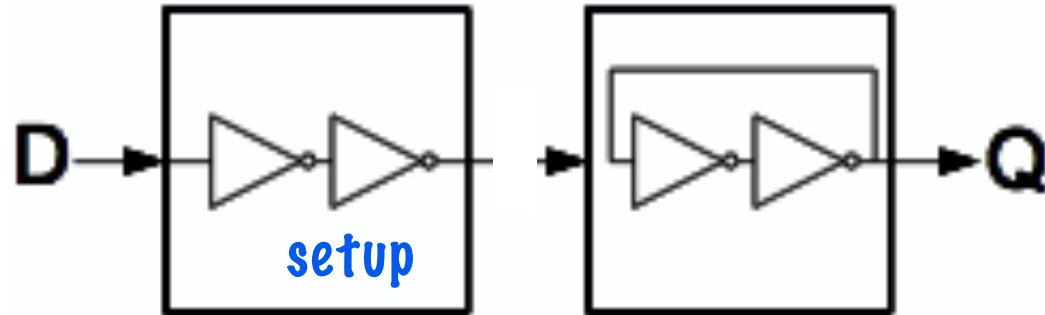
CLK 0->1

Capture D, pass value to Q

clk-to-Q
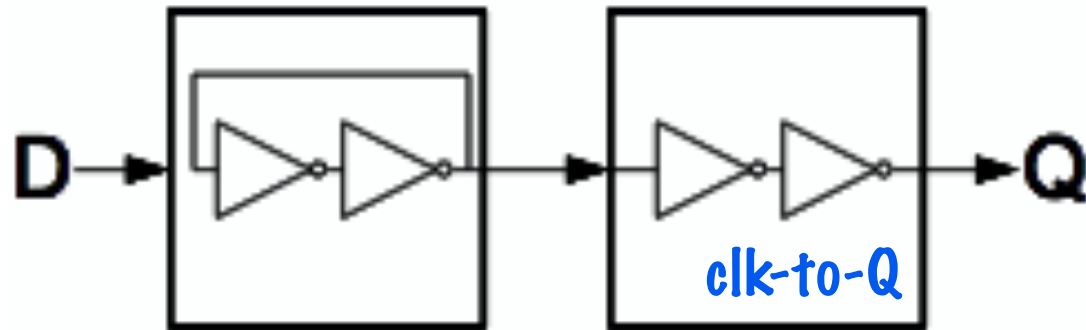
Note: with too much fanout, second stage could fail to capture data properly. Often output is rebuffered.

Hold-time Violations

clk

d — FF — q

clk

input must be stable

t_{set_up} "setup time"

d

t_{hold} "hold time"

q

$t_{clk\_to\_q}$ delay from clk to output
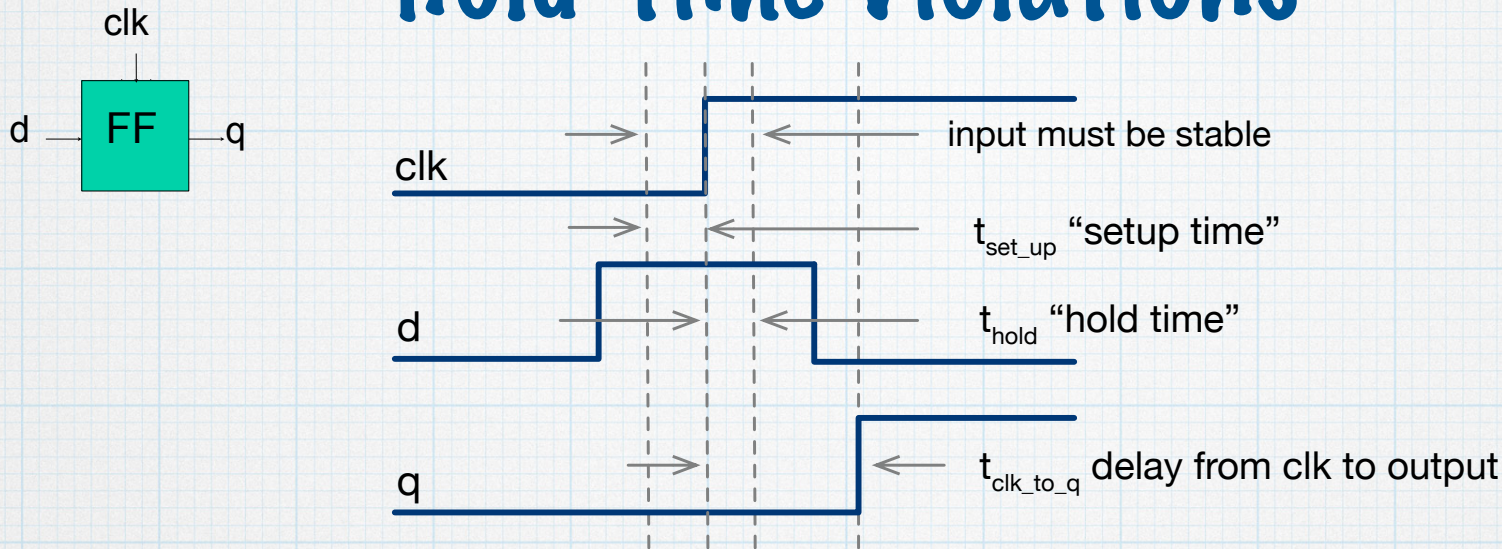
▸ Some state elements have positive hold time requirements.

  ▸ How can this be?

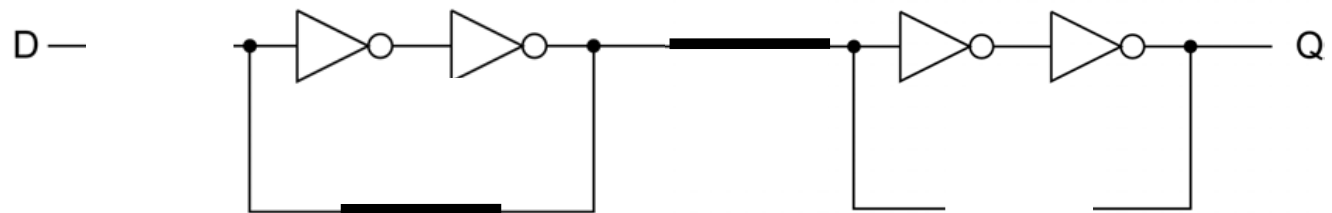▸ Fast paths from one state element to the next can create a violation. (Think about shift registers!)

▸ CAD tools do their best to fix violations by inserting delay (buffers).

  ▸ Of course, if the path is delayed too much, then cycle time suffers.

  ▸ Difficult because buffer insertion changes layout, which changes path delay.

Register

F1

F2

Instruction
Cache

instruction
word

ID
TH/ID

Thumb
Decode

ID

Combinational Logic

RF

RF

Shifter

ALU

X1

X2

Sat

the same clock

F Write
Port 1

WB

DWB
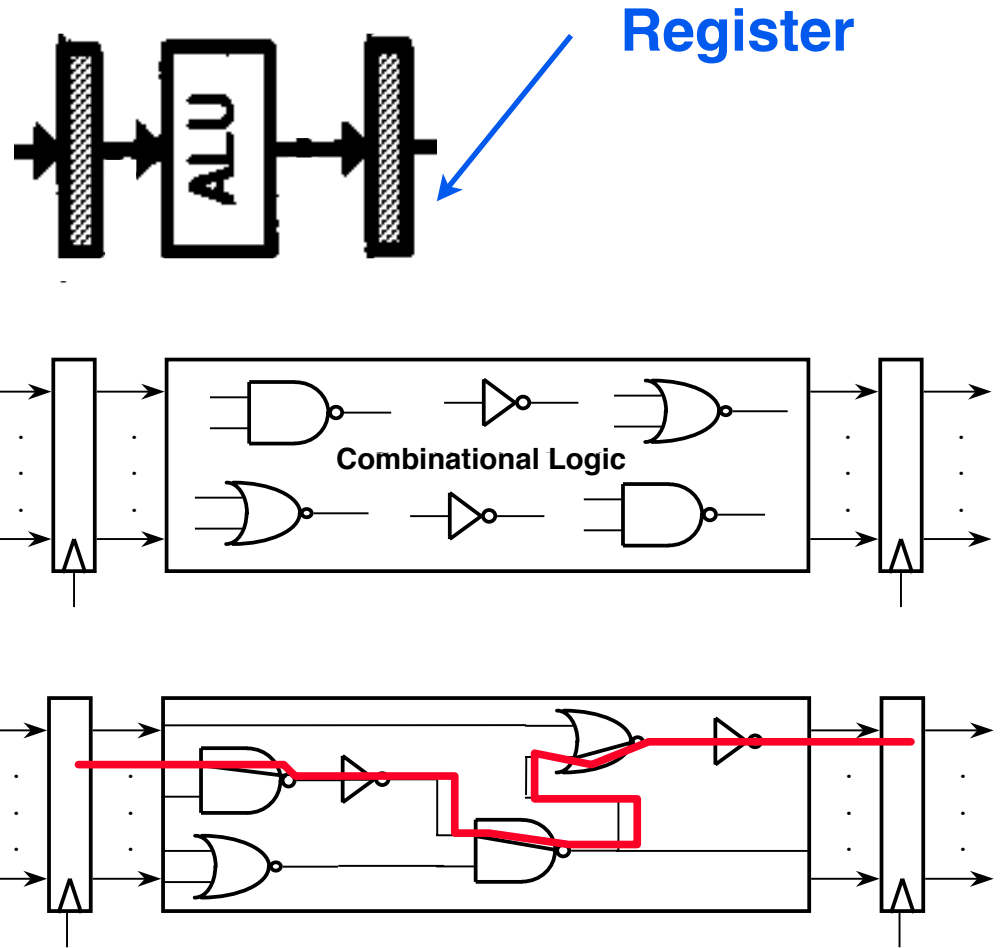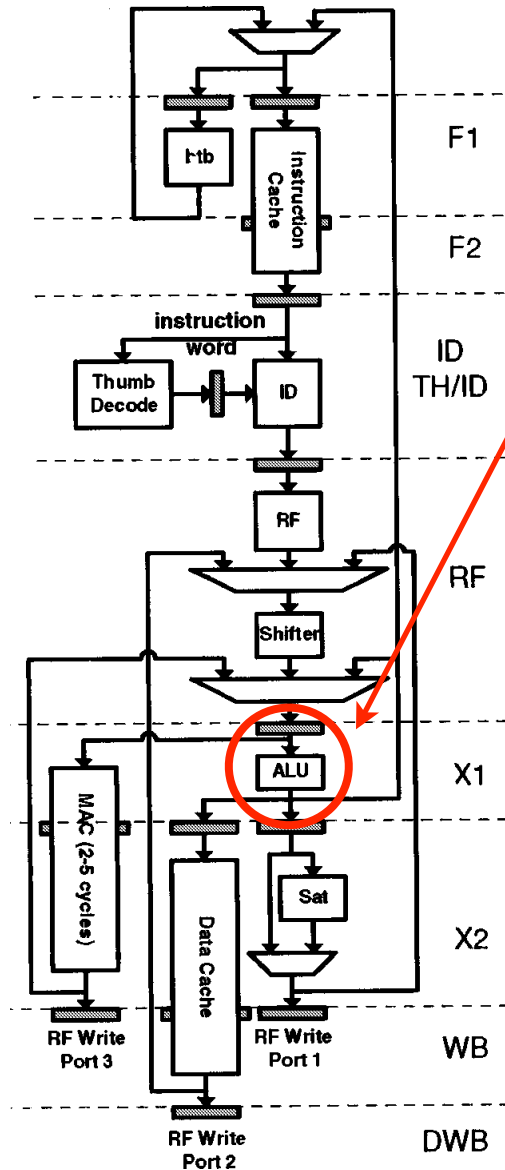
Some path somewhere in the design has the longest delay and is therefore the "critical path".

Components of Combinational Path Delay

CL ≡ combinational logic cell
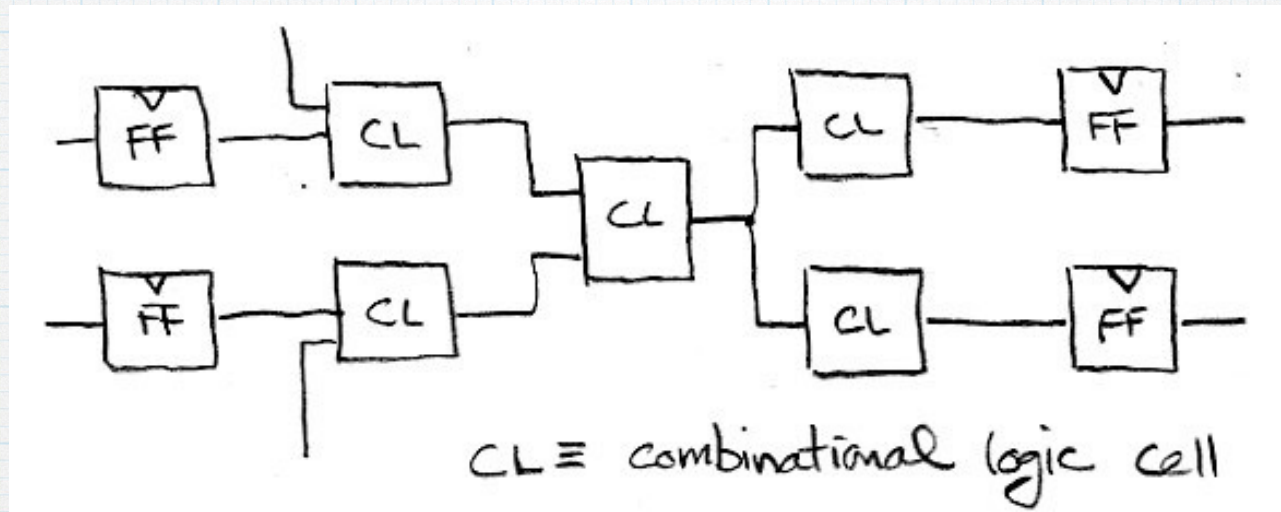
1. # of levels of logic
2. Internal cell delay
3. wire delay
4. cell input capacitance
5. cell fanout
6. cell output drive strength

Who controls the delay in ASIC?

	foundary engineer (TSMC)	Library Developer (Aritsan)	CAD Tools (DC, IC Compiler)	Designer (you!)
1. # of levels			synthesis	HDL design
2. Internal cell delay	physical parameters	cell topology, trans sizing	cell selection	
3. Wire delay	physical parameters		place & route	layout
4. Cell input capacitance	physical parameters	cell topology, trans sizing	cell selection	
5. Cell fanout			synthesis	HDL design
6. Cell drive strength	physical parameters	transistor sizing	cell selection	

Timing Closure: Searching for and beating down the critical path
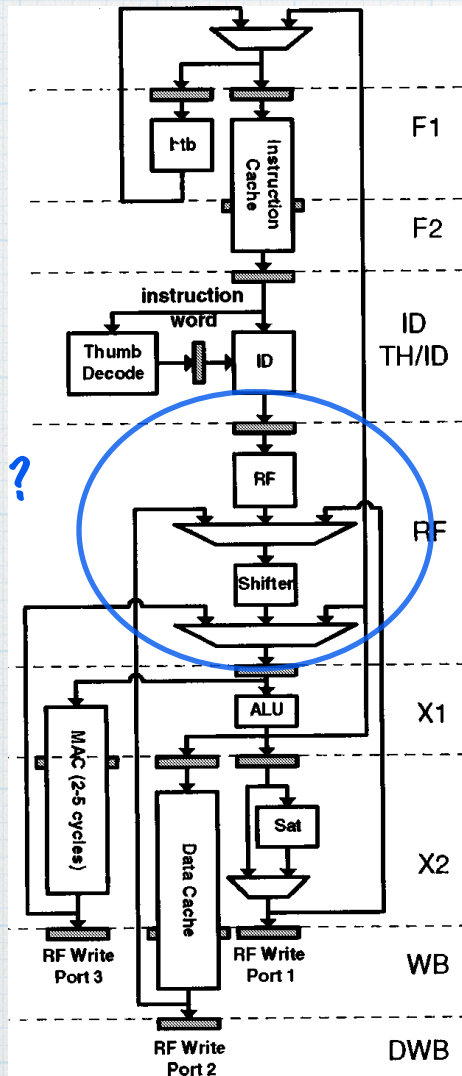
Must consider all connected register pairs, paths, plus from input to register, plus register to output.

- Design tools help in the search.

- Synthesis tools work to meet clock constraint, report delays on paths,

  - Special static timing analyzers accept a design netlist and report path delays,

  - and, of course, simulators can be used to determine timing performance.

Tools that are expected to do something about the timing behavior (such as synthesizers), also include provisions for specifying input arrival times (relative to the clock), and output requirements (set-up times of next stage).

Timing Analysis, real example

The critical path

Most paths have hundreds of picoseconds to spare.
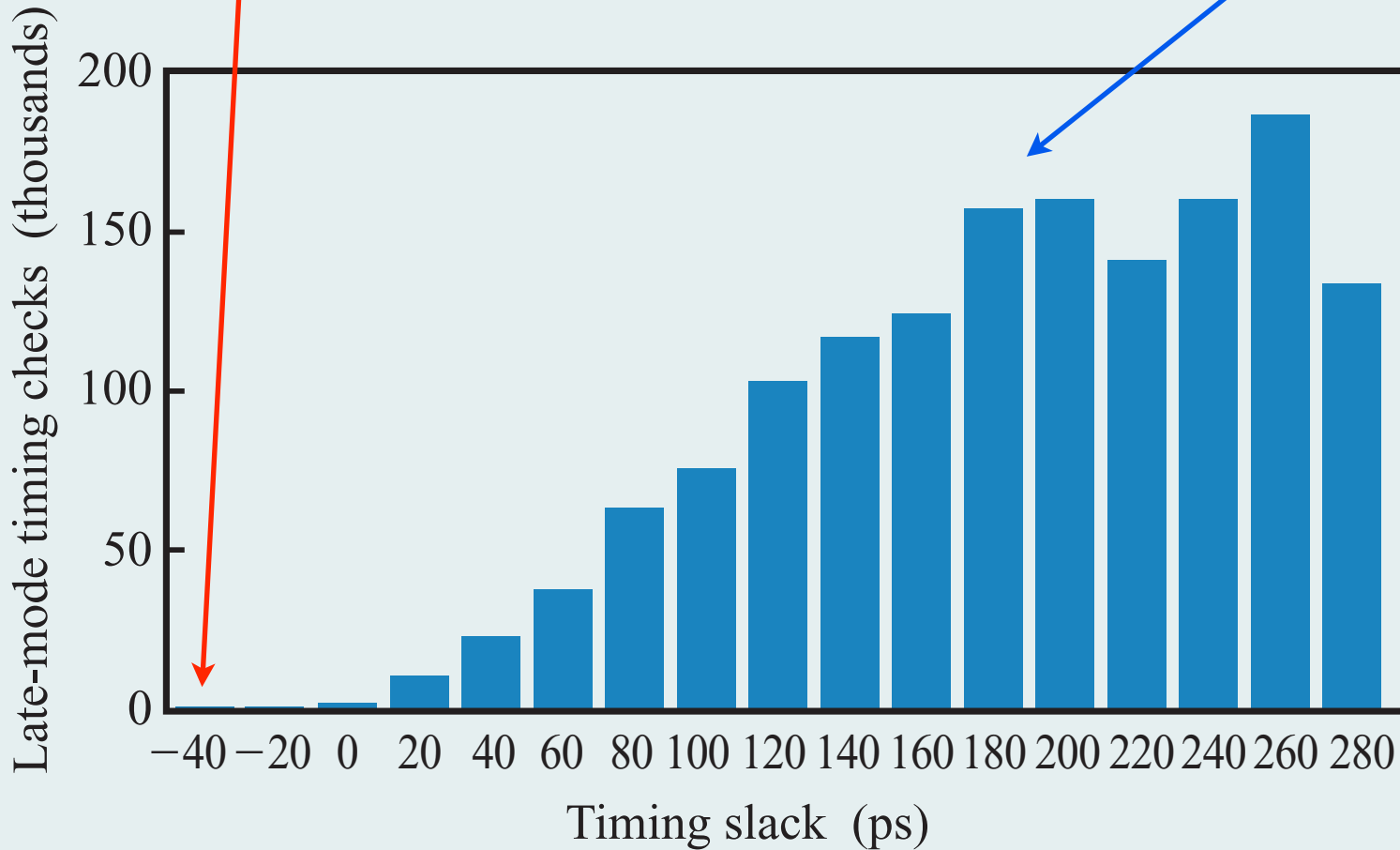
From "The circuit and physical design of the POWER4 microprocessor", IBM J Res and Dev, 46:1, Jan 2002, J.D. Warnock et al.

Timing Optimization

As an ASIC/FPGA designer you get to choose:

- ▶ The algorithm

- ▶ The Microarchitecture (block diagram)

- ▶ The HDL description of the CL blocks (number of levels of logic)

- ▶ Where to place registers and memory (the pipelining)

- ▶ Overall floorplan and relative placement of blocks

Circuit retiming*

Circles are combinational logic, labelled with delays.

Critical path is 5 (ignore FF delay for now).

We want to improve it without changing circuit semantics.

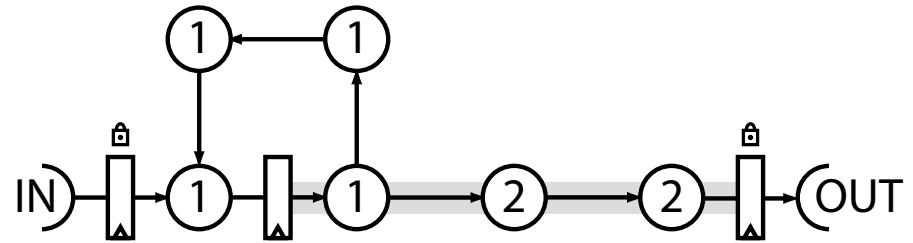Figure 1: A small graph before retiming. The nodes represent logic delays, with the inputs and outputs passing through mandatory, fixed registers. The critical path is 5.

Add a register, move one circle.

Performance improves by 20%.
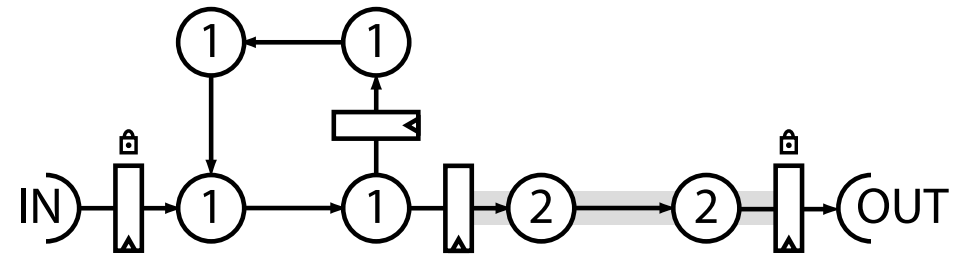
Figure 2: The example in Figure 2 after retiming. The critical path is reduced from 5 to 4.

*a.k.a. Register Rebalancing

Logic Synthesis tools can do this in simple cases.

Retiming Example

Clk->Q
10

Clk->Q
10

30

REG

30

20

REG

30

30

REG

REG

Setup
20

$\tau_{Clk->Q} = 10$

$\tau_{Setup} = 20$

Critical Path Delay = 190
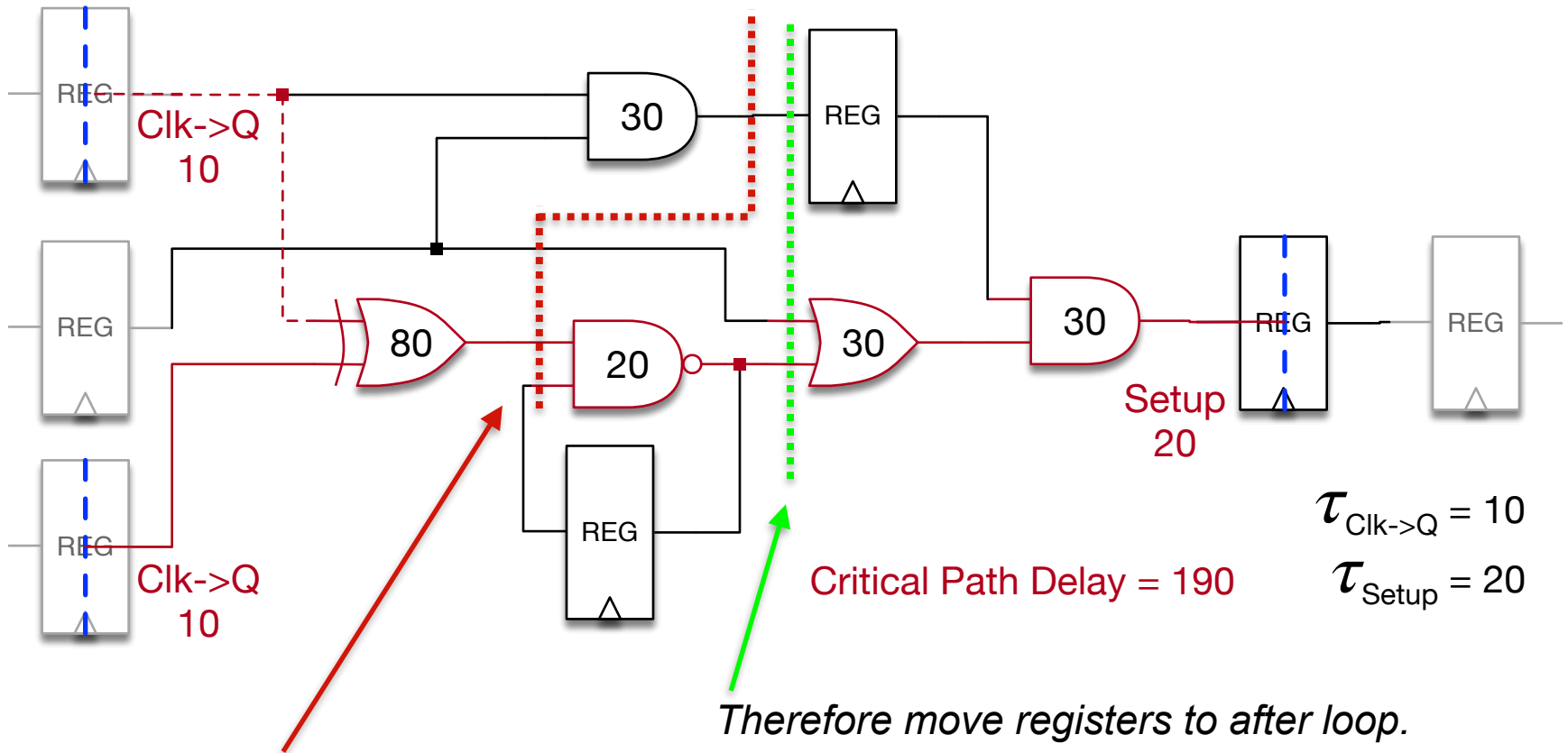
Want to retime to here, however, delay cannot be added to the loop without changing the semantics of the logic. Because of this, many retiming tools stop at loops.

Therefore move registers to after loop.

Retiming Example

Clk->Q
10

REG

REG

Clk->Q
10

30

80

20

REG

REG

REG

REG

Setup
20

Setup
20

30

30

REG

Critical Path Delay
= 130

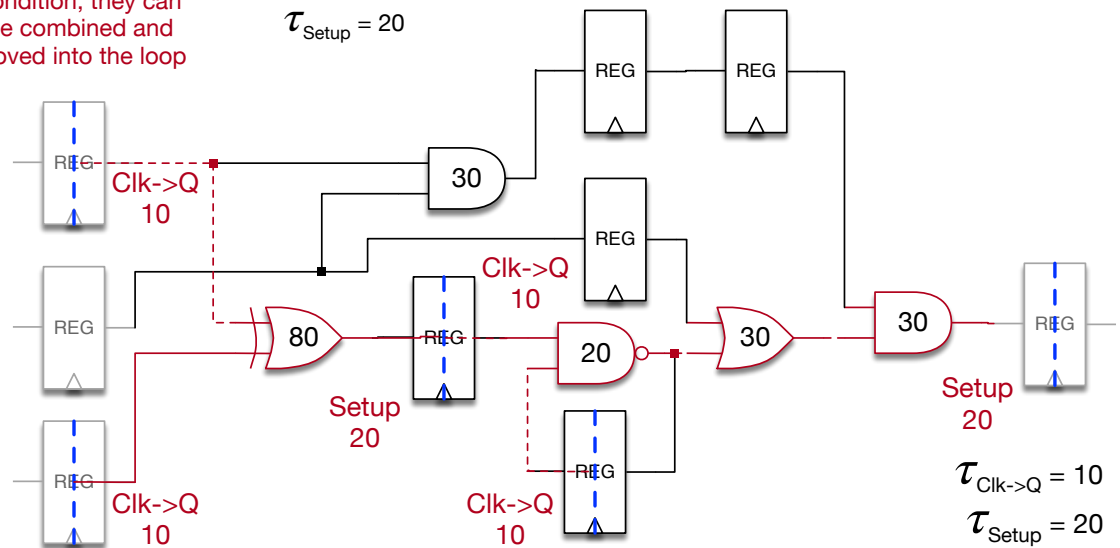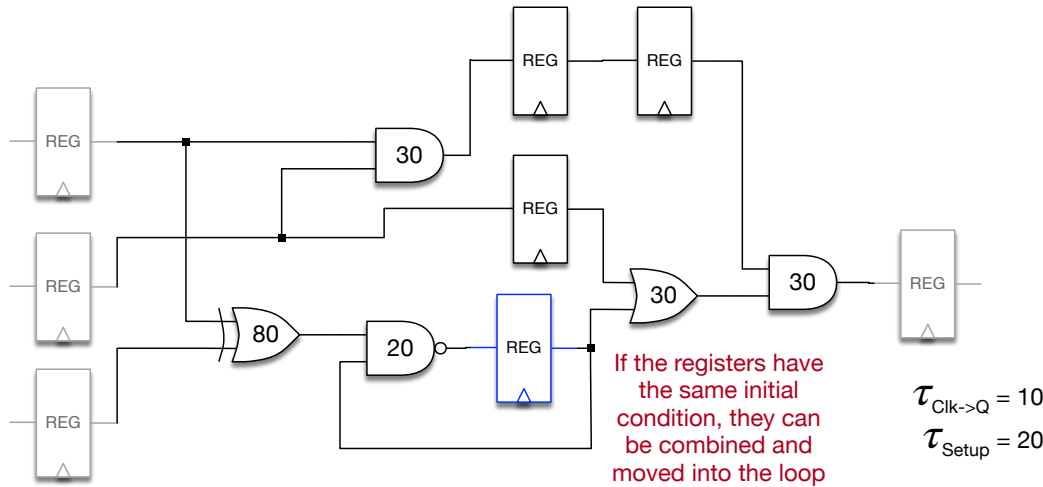$\tau_{Clk->Q} = 10$

$\tau_{Setup} = 20$

This is the retimed solution that
many retiming aware tools will stop at.

*This is also the optimal solution when
the initial values of the registers are not given.*

If the registers have
the same initial
condition, they can
be combined and
moved into the loop

Retiming Example

If the registers have the same initial condition, they can be combined and moved into the loop

$\tau_{\text{Clk->Q}} = 10$

$\tau_{\text{Setup}} = 20$

Clk->Q
10

Clk->Q
10

Setup
20

Clk->Q
10

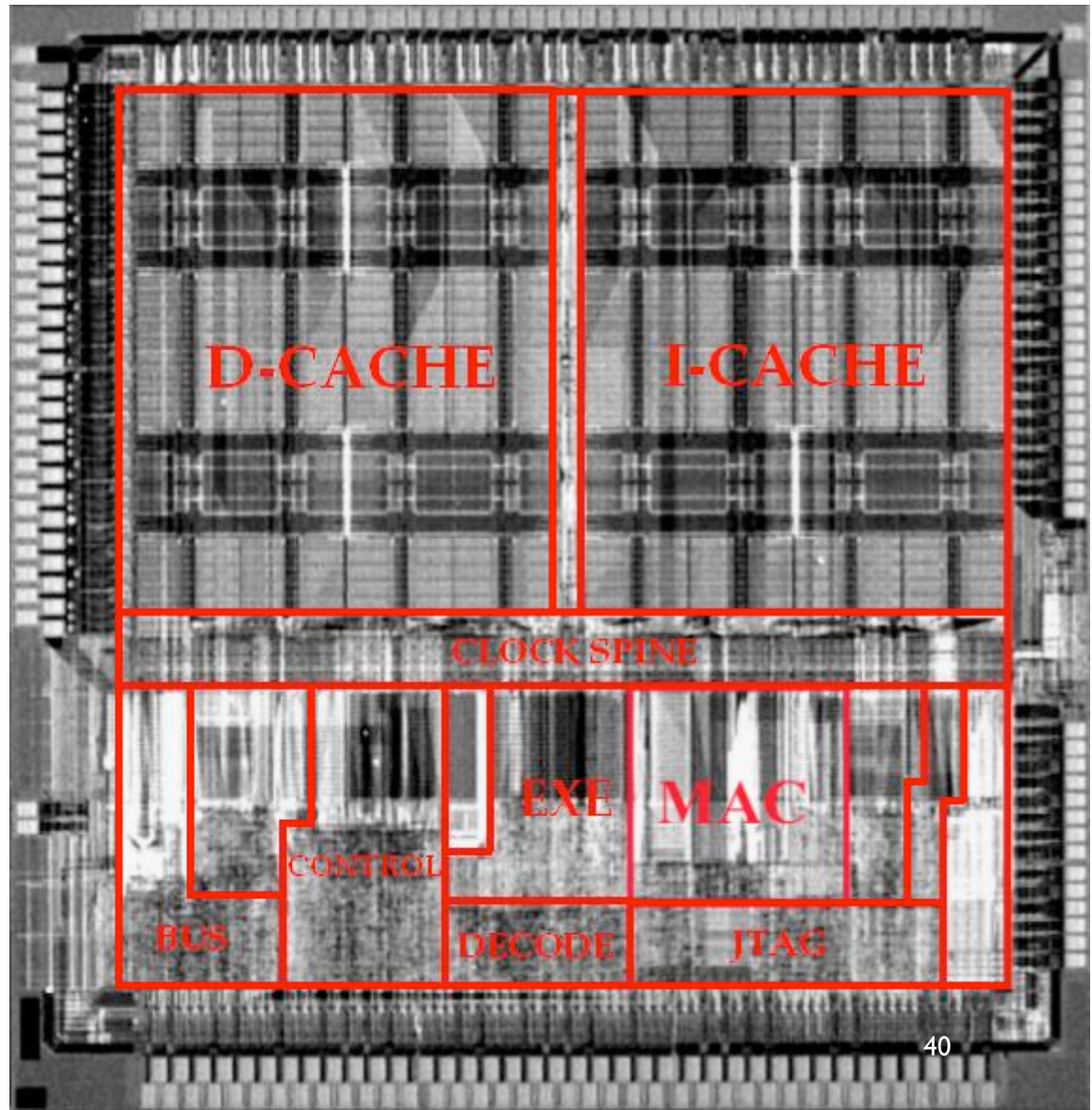Clk->Q
10

Setup
20

$\tau_{\text{Clk->Q}} = 10$

$\tau_{\text{Setup}} = 20$
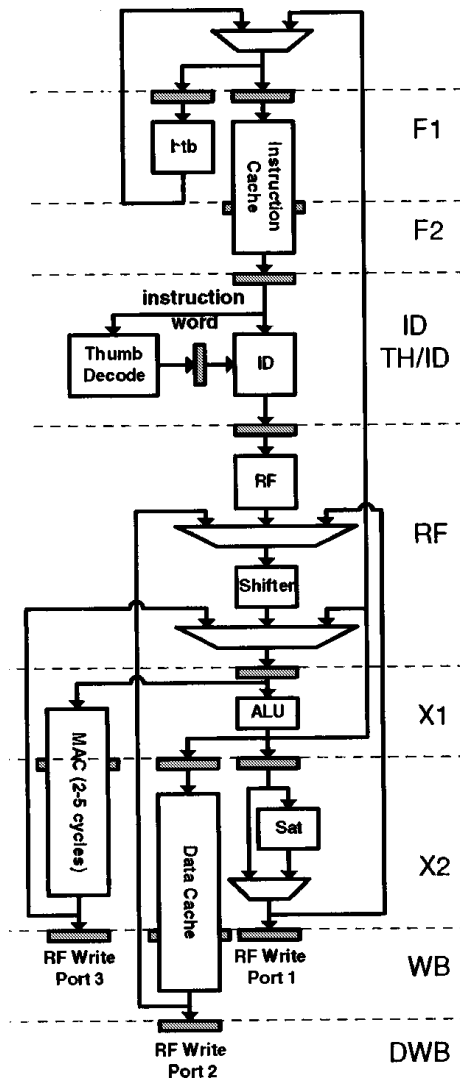
The register can be moved through the NAND gate, producing, a register at each input. The total delay in the loop is unchanged. Care must be taken to properly set the initial conditions of the registers.

Critical Path Delay
= 110

Floorplaning: essential to meet timing.

(Intel XScale 80200)

Timing Analysis Tools

- Static Timing Analysis: Tools use delay models for gates and interconnect. Traces through circuit paths.
  - Cell delay model capture
   - For each input/output pair, internal delay (output load independent)
    - output dependent delay
- Standalone tools (PrimeTime) and part of logic synthesis.
- Back-annotation takes information from results of place and route to improve accuracy of timing analysis.
- DC in "topographical mode" uses preliminary layout information to model interconnect parasitics.
  - Prior versions used a simple fan-out model of gate loading.

delay

output load
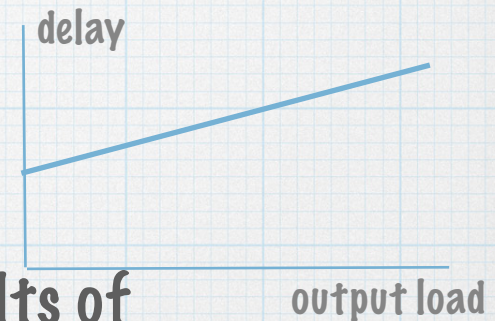
Standard cell characterization
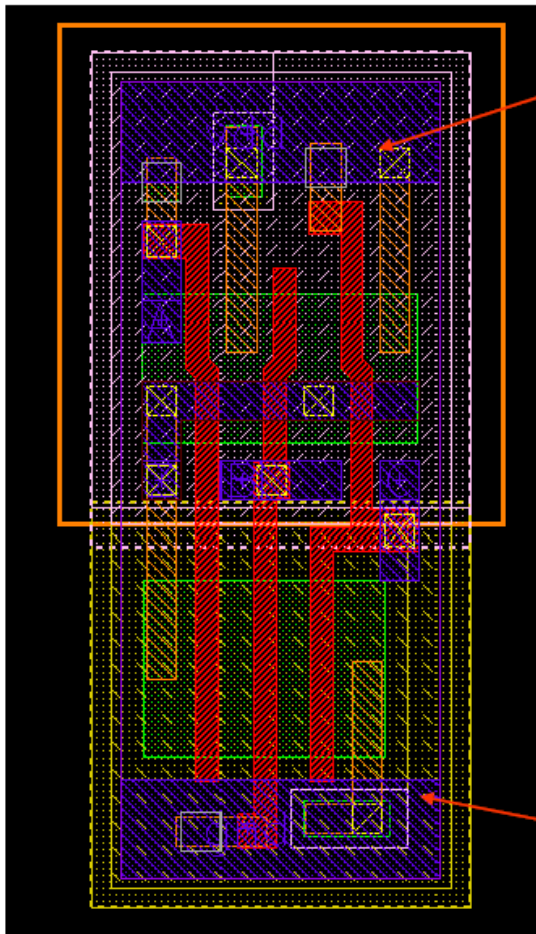
Power Supply Line (V_{DD})   **Delay in (ns)!!**

Path	1.2V - 125°C	1.6V - 40°C
$In1$—t_{pLH}	$0.073+7.98C+0.317T$	$0.020+2.73C+0.253T$
$In1$—t_{pHL}	$0.069+8.43C+0.364T$	$0.018+2.14C+0.292T$
$In2$—t_{pLH}	$0.101+7.97C+0.318T$	$0.026+2.38C+0.255T$
$In2$—t_{pHL}	$0.097+8.42C+0.325T$	$0.023+2.14C+0.269T$
$In3$—t_{pLH}	$0.120+8.00C+0.318T$	$0.031+2.37C+0.258T$
$In3$—t_{pHL}	$0.110+8.41C+0.280T$	$0.027+2.15C+0.223T$

3-input NAND cell
(from ST Microelectronics):
C = Load capacitance
T = input rise/fall time

Ground Supply Line (GND)

- Each library cell (FF, NAND, NOR, INV, etc.) and the variations on size (strength of the gate) is fully characterized across temperature, loading, etc.

End of Lecture 11