# EECS150 - Digital Design
# Lecture 19 –CMOS Implementation Technologies

Oct. 31, 2013
Prof. Ronald Fearing
Electrical Engineering and Computer Sciences
University of California, Berkeley

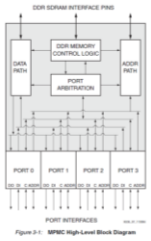
(slides courtesy of Prof. John Wawrzynek)

http://www-inst.eecs.berkeley.edu/~cs150

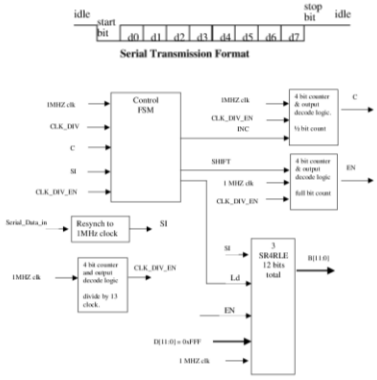


## Recap and Outline

Arbiter



Univ. Async. Receiver




A/D Converter

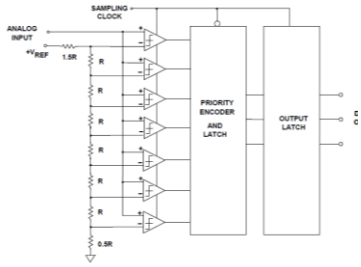



Figure 4: 3-bit All-Parallel (Flash) Converter

### Outline for Today

- Midterm 1 Feedback
  - regrade request with note to RSF by Tues Nov. 5, 5 pm.
- CMOS details

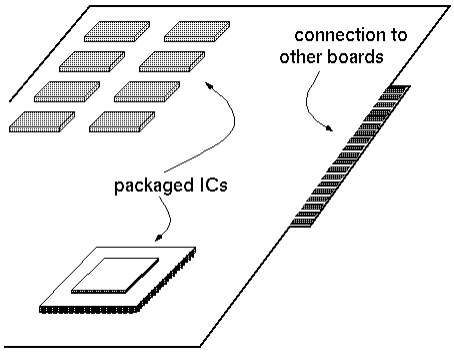
# Overview of Physical Implementations

*The stuff out of which we make systems.*

- Integrated Circuits (ICs)
  - Combinational logic circuits, memory elements, analog interfaces.
- Printed Circuits (PC) boards
  - substrate for ICs and interconnection, distribution of CLK, Vdd, and GND signals, heat dissipation.
- Power Supplies
  - Converts line AC voltage to regulated DC low voltage levels.
- Chassis (rack, card case, ...)
  - holds boards, power supply, fans, provides physical interface to user or other systems.
- Connectors and Cables.



# Printed Circuit Boards

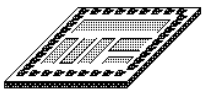



- fiberglass or ceramic
- 1-25 conductive layers
- ~1-20in on a side
- IC packages are soldered down.

## Multichip Modules (MCMs)

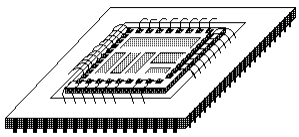
- Multiple chips directly connected to a substrate. (silicon, ceramic, plastic, fiberglass) without chip packages.

# Integrated Circuits



- Primarily Crystalline Silicon
- 1mm - 25mm on a side
- 100 - 1000M transistors
- (25 - 250M “logic gates")
- 3 - 10 conductive layers
- 2012 - feature size ~ 28nm = 0.028 x $10^{-6}$ m
- “CMOS” most common - complementary metal oxide semiconductor

## Chip in Package

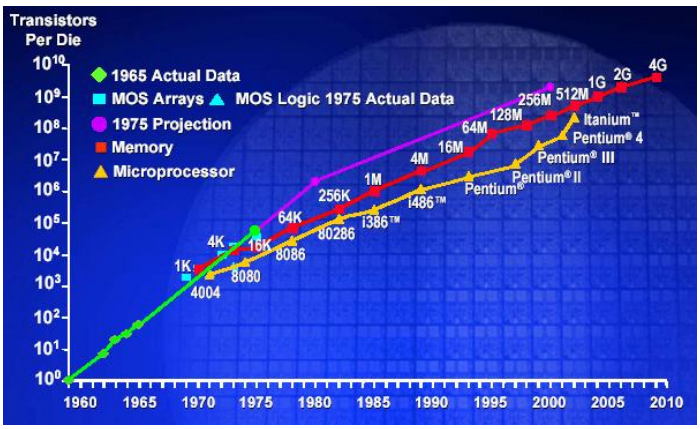


- Package provides:
  - spreading of chip-level signal paths to board-level
  - heat dissipation.
- Ceramic or plastic with gold wires.



# Integrated Circuits

- Moore’s Law has fueled innovation for the last 3 decades.




- “Number of transistors on a die doubles every 18 months.”
- What are the consequences of Moore’s law?

# Chip-level Function Implementation Alternatives

Full-custom: All circuits/transistor layouts optimized for application.

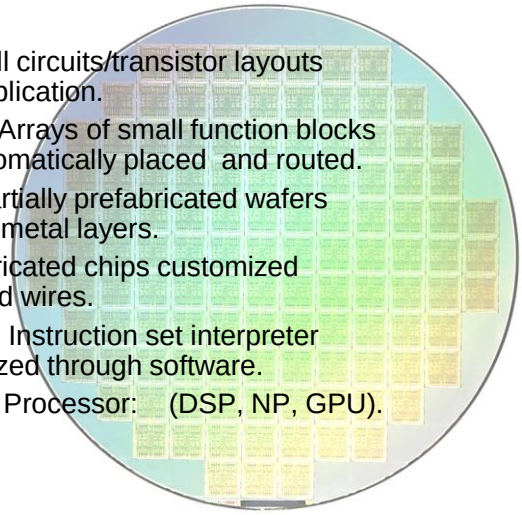
Standard-cell: Arrays of small function blocks (gates, FFs) automatically placed and routed.

Gate-array: Partially prefabricated wafers customized with metal layers.

FPGA: Prefabricated chips customized with switches and wires.

Microprocessor: Instruction set interpreter customized through software.

Domain Specific Processor: (DSP, NP, GPU).



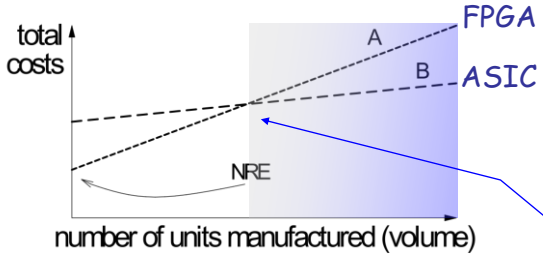
**ASIC**

What are the important metrics of comparison?



# Why FPGAs?

A tradeoff exists between NRE* cost and manufacturing costs:




The ASIC approach is only viable for products with very high volume (where NRE could be amortized), and which were not time to market (TTM) sensitive.

Cross-over point has moved to the right (favoring FPGA) implementation as ASIC NREs have increased.

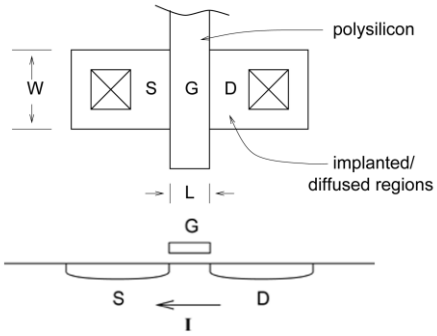
*Non-recurring Engineering Costs

# CMOS Devices

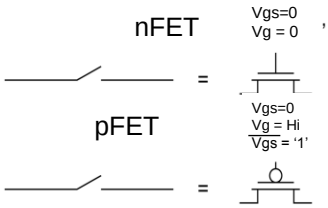
- MOSFET (Metal Oxide Semiconductor Field Effect Transistor).

*Top View*

*Cross Section*



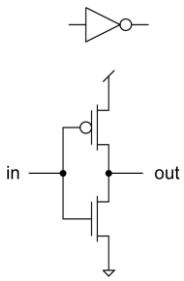

The gate acts like a capacitor. A high voltage on the gate attracts charge into the channel. If a voltage exists between the source and drain a current will flow. In its simplest approximation, the device acts like a switch.



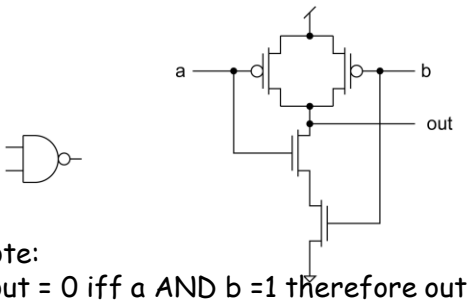



# Transistor-level Logic Circuits

Inverter (NOT gate):




NAND gate:




Note:

- out = 0 iff a AND b =1 therefore out = (ab)'
- pFET network and nFET networks are duals of one another.

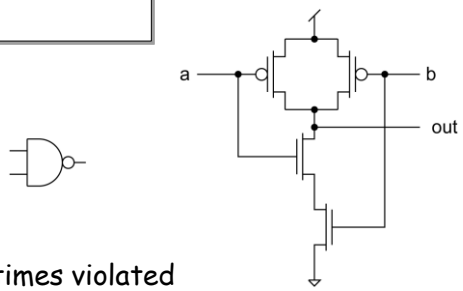
How about AND gate?

# Transistor-level Logic Circuits

Simple rule for wiring up MOSFETs:

nFET is used only to pass logic zero.
pFET is used only to pass logic one.

For example, consider the NAND gate:



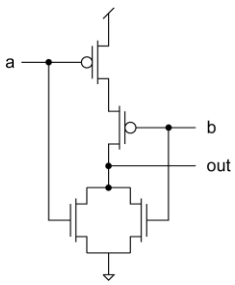



Note: This rule is sometimes violated by expert designers under special conditions.



# Transistor-level Logic Circuits

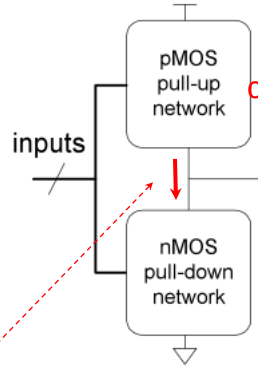
NOR gate:




Note:
- out = 0 iff a OR b =1 therefore out = (a+b)'
- Again pFET network and nFET networks are duals of one another.

Other more complex functions are possible. Ex: out = (a+bc)'

# CMOS Logic Gates in General




Pull-up network conducts under conditions to generate a logic 1 output

Pull-down network conducts for logic 0 output

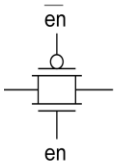
Conductance must be mutually exclusive - else, short circuit!

Pull-up and pull-down networks are "topological duals"



# Transmission Gate

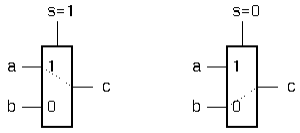
- Transmission gates are the way to build "switches" in CMOS.
- In general, both transistor types are needed:
  - nFET to pass zeros.
  - pFET to pass ones.
- The transmission gate is bi-directional (unlike logic gates).



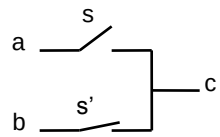

- Does not directly connect to Vdd and GND, but can be combined with logic gates or buffers to simplify many logic structures.

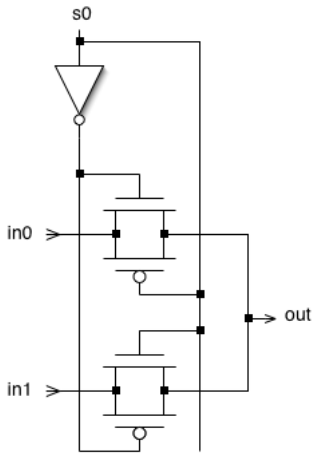
# Transmission-gate Multiplexor

2-to-multiplexor:
$C = sa + s'b$




Switches simplify the implementation:

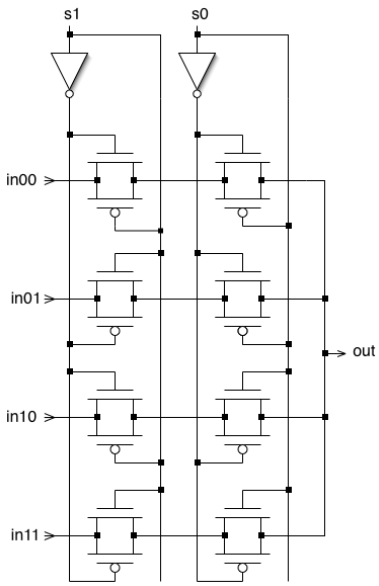



Compare the cost to logic gate implementation.



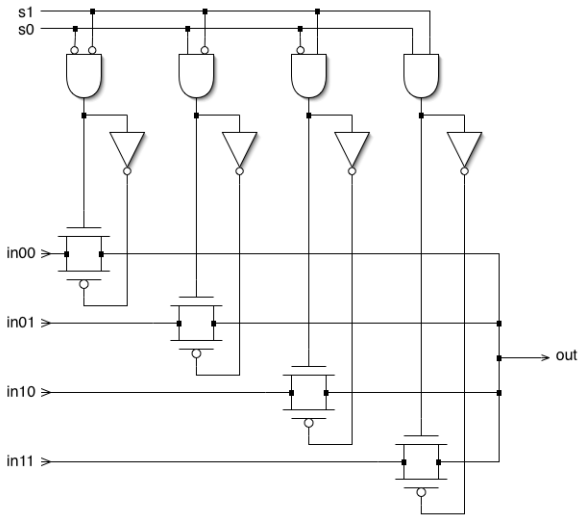
# 4-to-1 Transmission-gate Mux




- The series connection of pass-transistors in each branch effectively forms the AND of s1 and s0 (or their complement).
- Compare cost to logic gate implementation

# Alternative 4-to-1 Multiplexor

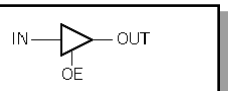
- This version has less delay from in to out.
- In both versions, care must be taken to avoid turning on multiple paths simultaneously (shorting together the inputs).






# Tri-state Buffers

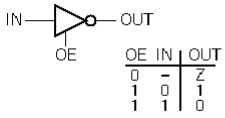
Tri-state Buffer:




| OE | IN | OUT |
|---|---|---|
| 0 | 0 | Z |
| 0 | 1 | Z |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

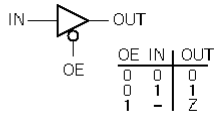
"high impedance"
(output disconnected)

Variations:



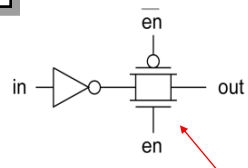

| OE | IN | OUT |
|---|---|---|
| 0 | – | Z |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

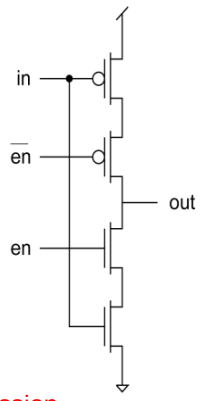
Inverting buffer




| OE | IN | OUT |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | – | Z |

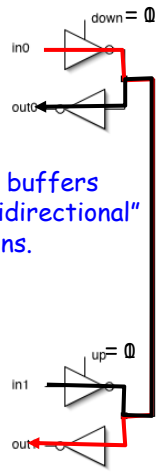
Inverted enable



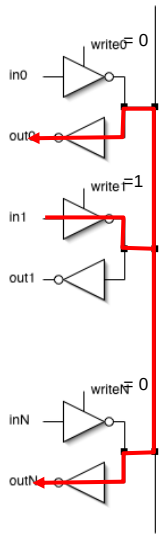

transmission gate useful in implementation

# Tri-state Buffers




Tri-state buffers enable "bidirectional" connections.

Tri-state buffers are used when multiple circuits all connect to a common wire. Only one circuit at a time is allowed to drive the bus. All others "disconnect" their outputs, but can "listen".

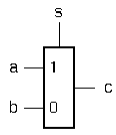




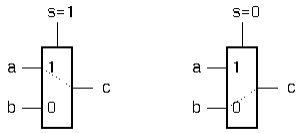

= 0

# Tri-state Based Multiplexor

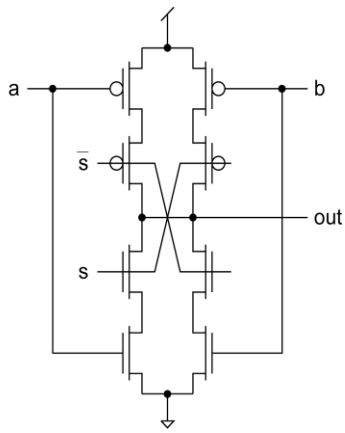
Multiplexor




If s=1 then c=a

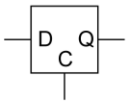


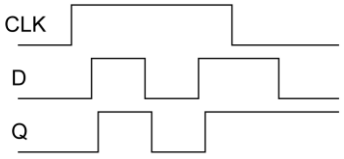

Transistor Circuit for inverting multiplexor:



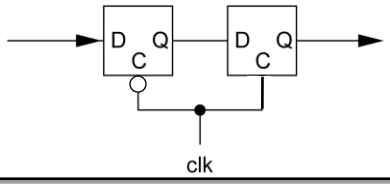
# Latches and Flip-flops

Positive level-sensitive **latch**:

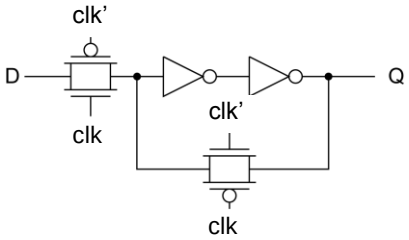



Positive Edge-triggered **flip-flop** built from two level-sensitive latches:




**Latch** Implementation:

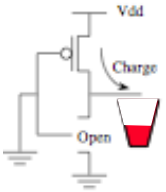


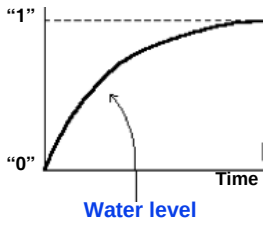



# CMOS Delay: Transistors as water valves

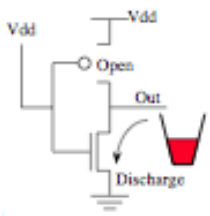
**If electrons are water molecules, and a capacitor a bucket ...**

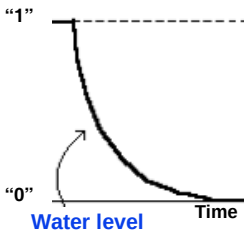
**A "on" p-FET fills up the capacitor with charge.**




**A "on" n-FET empties the bucket.**



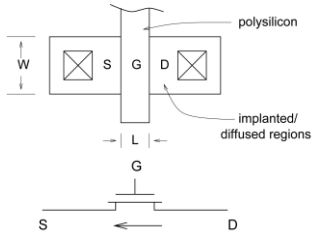






This model is often good enough

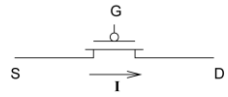
# Transistors as Conductors

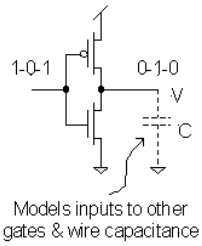
- Improved Transistor Model:
  **nFET**

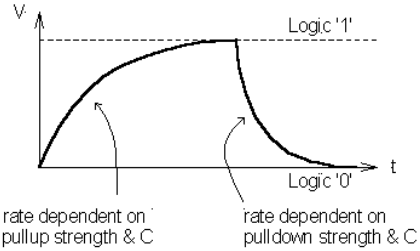



- We refer to transistor "strength" as the amount of current that flows for a given Vds and Vgs.
- The strength is linearly proportional to the ratio of W/L.

**pFET**



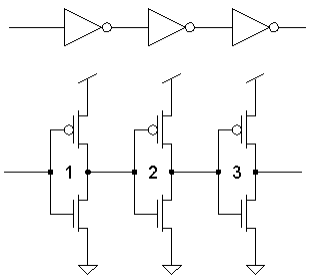



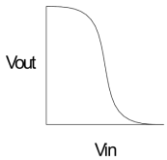





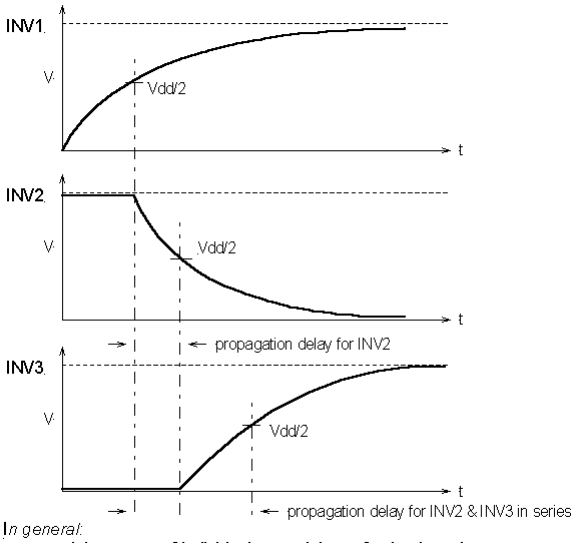

# Gate Delay is the Result of Cascading

- Cascaded gates:



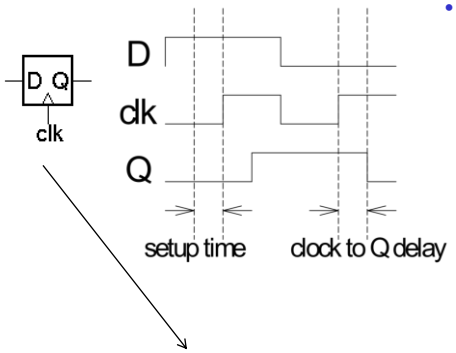




"transfer curve" for inverter.

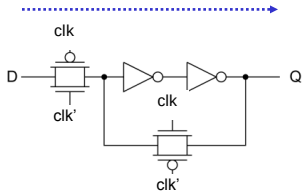



*In general:*
**prop. delay = sum of individual prop. delays of gates in series.**

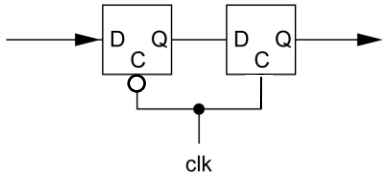
# Delay in Flip-flops

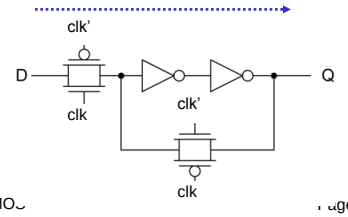



- Setup time results from delay through *first* latch.



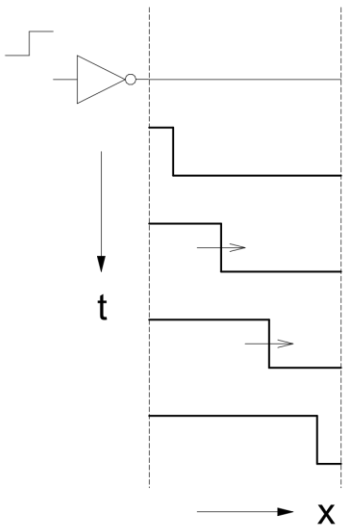

Clock to Q delay results from delay through *second* latch.





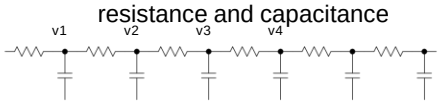

# Wire Delay




- Ideally, wires behave as "transmission lines":
  - signal wave-front moves close to the speed of light
    - ~1ft/ns
  - Time from source to destination is called the "transit time".
  - In ICs most wires are short, and the transit times are relatively short compared to the clock period and can be ignored.
  - Not so on PC boards.

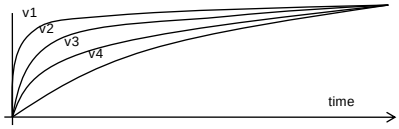
# Wire Delay

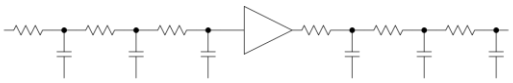
- Even in those cases where the transmission line effect is negligible:
  - Wires posses distributed resistance and capacitance




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

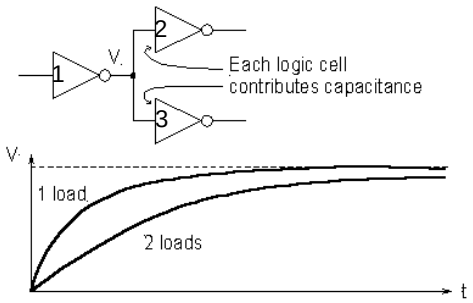
- For **short wires** on ICs, resistance is insignificant (relative to effective R of transistors), but C is important.
  - Typically around half of C of gate load is in the wires.
- For **long wires** on ICs:
  - busses, clock lines, global control signal, etc.
  - Resistance is significant, therefore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





# Delay and "Fan-out"




- The delay of a gate is proportional to its output capacitance. Connecting the output of gate to more than one other gate increases it's output capacitance. It takes increasingly longer for the output of a gate to reach the switching threshold of the gates it drives as we add more output connections.
- Driving wires also contributes to fan-out delay.
- What can be done to remedy this problem in large fan-out situations?

# Conclusions

- CMOS details

29