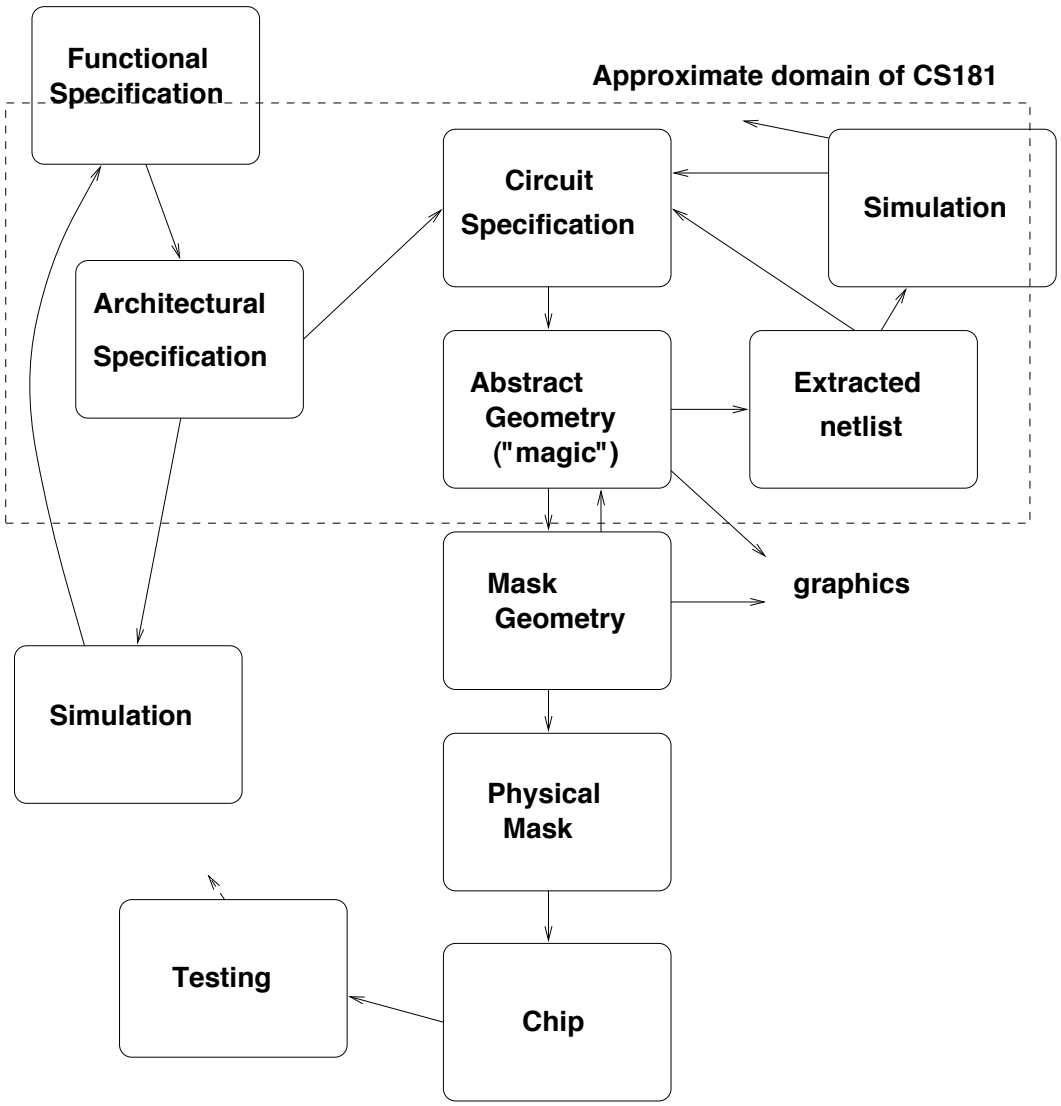
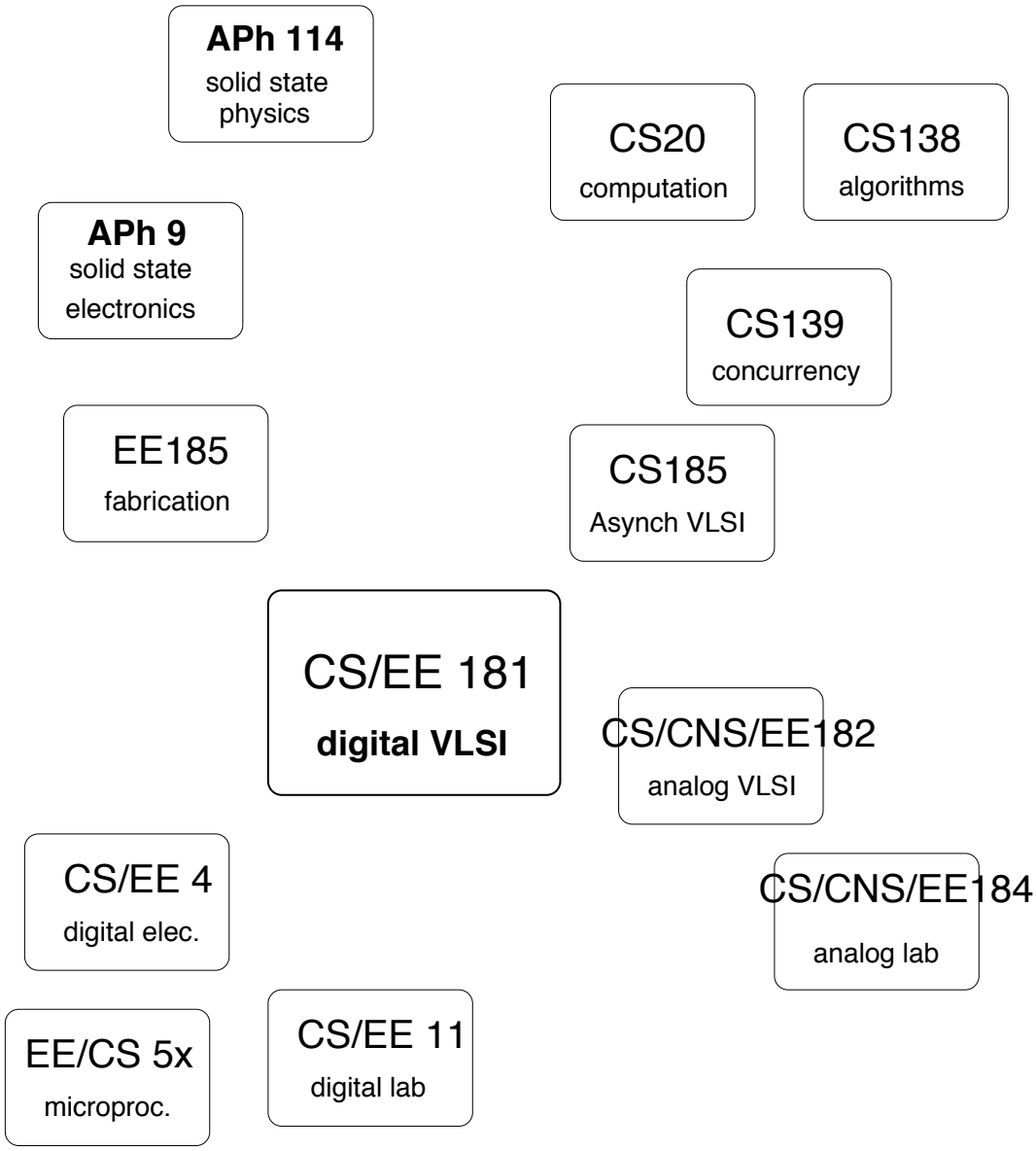
CS/EE 181a 2013/14 Lecture 1

CS/EE 181 is about designing digital CMOS systems.



Related Courses



Is there method to this madness? Or does the designer have to be an expert in all these areas?

Why CMOS?

All we really care about is implementing algorithms in hardware—so why digital, synchronous CMOS?

There used to be many alternatives:

- TTL, NMOS, ECL...
- Ge, GaAs...
- currents instead of voltages(?)
- Analog design
- A few new ones (?): Optical, superconducting, quantum

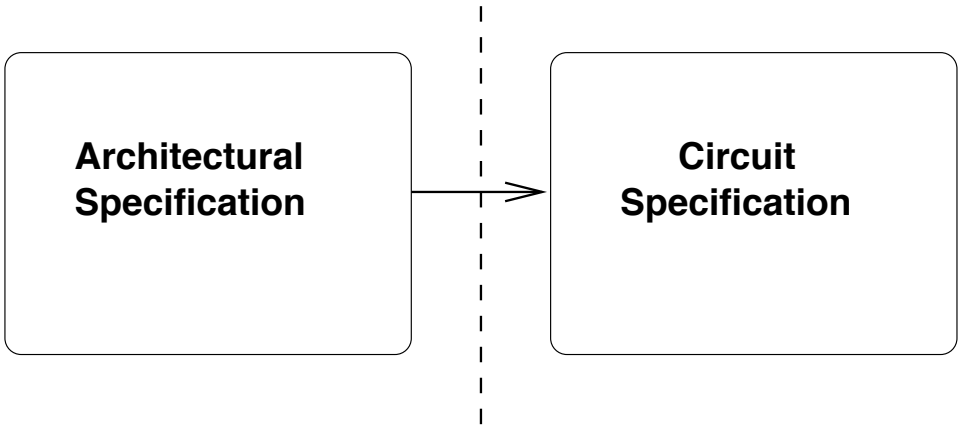
Goals:

- Correctness (most important)
- High performance–speed and/or power (almost as important)

CMOS is ideal for achieving our goals with reasonable effort. And computer scientists love it because it is (was?) easy to "program." But things are getting more difficult as scaling continues...

Abstraction

In CMOS, it is not difficult to translate "basically any logic equation" into reasonably efficient hardware.



Permits efficient design methodologies that generate circuits "correct by construction."

But human designers can still intervene & generate

- Faster, lower-power (i.e., better) circuits
- More errors. . .

Scaling

What happens to performance when we scale down the design?

Parameter	Was	Now	What happened
scale factor	1	$1/\alpha$	
linear size	x	x/α	Reduce dimensions
voltage	V	V/α	Reduce voltages
E-field	$\mathcal{E}$	$\mathcal{E}$	constant E-field
current	i	i/α	Reduce currents
power	vi	vi/α^2	(per device)
delay	τ	τ/α	Reduce delays
energy	$vi\tau$	$vi\tau/\alpha^3$	!

- Constant E-field scaling.

  The circuits get α faster with α^3 less energy. But they still work the same!

  Plus ça change, plus c'est la même chose.

Industrial concerns long kept voltages fixed—constant-voltage scaling. Now voltage scaling is slowing down again...more later.

Is everything wonderful...?

No...!

Problem with long wires

Delay in short wires: **drift** of charge carriers in electrical field:

$$t \sim L/\mathcal{E}$$

scales with α.

Delay in long wires: **diffusion** of charge carriers:

$$t \sim RCL^2$$

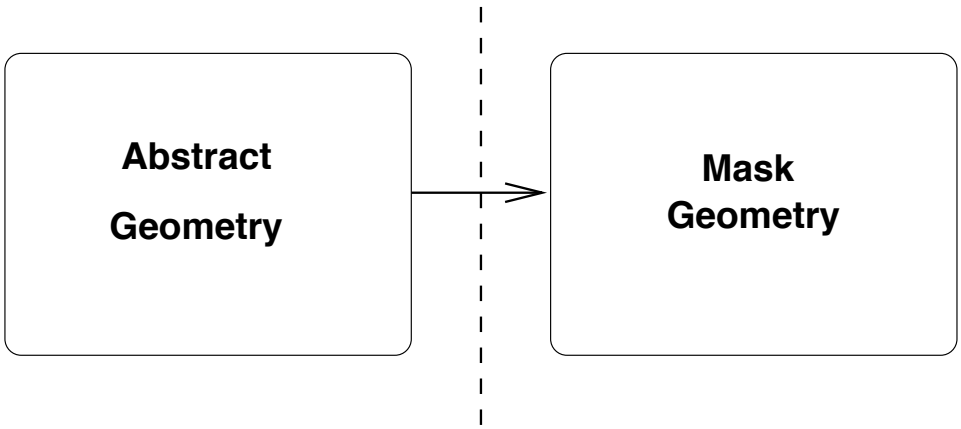
t remains constant with scaling.

Bad news! Long wires become relatively slower...and more capacitive.

Problem with power and energy consumption

More bad news! Increased PVT variability

More Abstraction

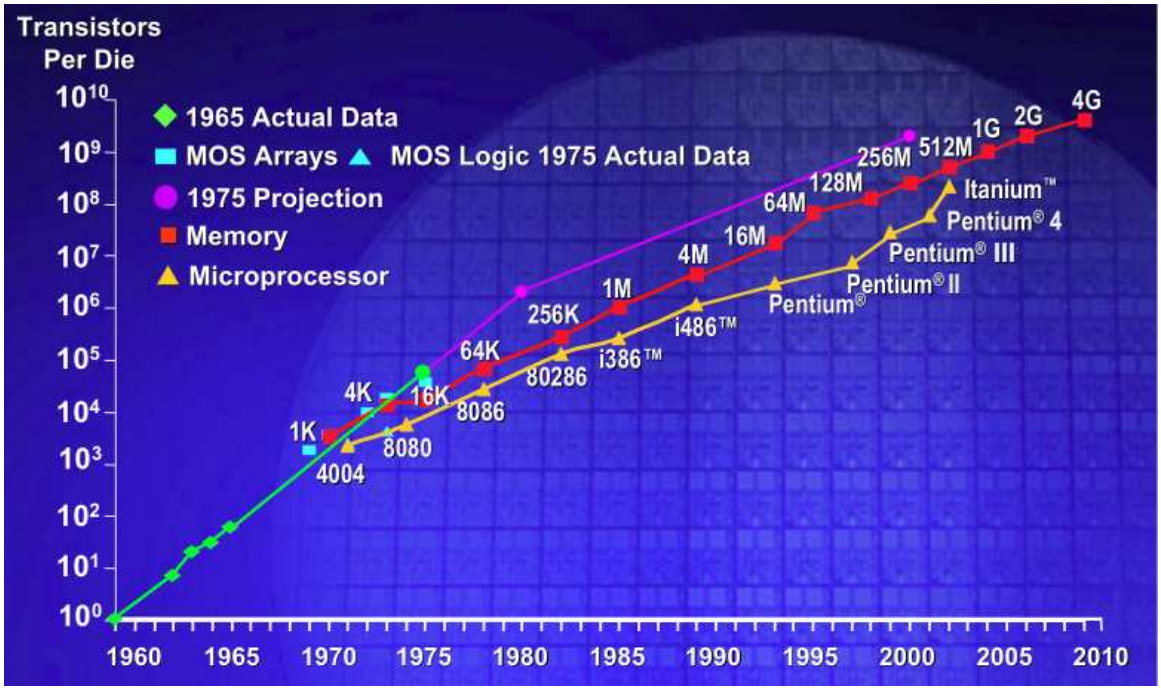


In `magic`, all our dimensions are in "λ." The CMOS scaling properties allow us to abstract away the actual physical dimensions.

- We can design a circuit once and refabricate it (ideally without functional verification) in a more modern technology...
- We can *think* about our systems the same way. So a 2.0-μm Tinychip has more than it might seem in common with a modern 1.5 billion-transistor microprocessor (Intel Ivy Bridge).

Trends

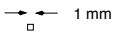
Scaling has been pursued aggressively since 1970. For the leading chips:

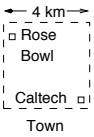


- *Moore's Law.* The number of transistors doubles every 18 months. . .or so. (Moore 1979.)

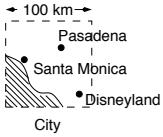
Imagine a city with each street as a wire on our chip, 200 m between blocks. (Seitz and Mead, 1979.)

1963
W=50 μm

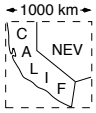




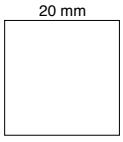
1975
W=10 μm



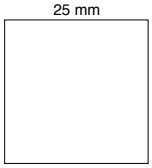
1985
W=2 μm



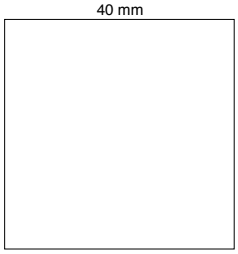
1995
W=.5 μm

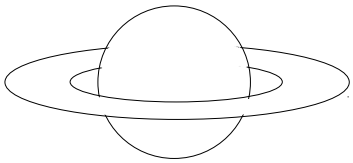


2000
W=.2 μm



2010??
W=.05 μm





Where we are today

There has been remarkable progress in microelectronics since 1960. Moore's Law has held through the entire period. (How long now...?)

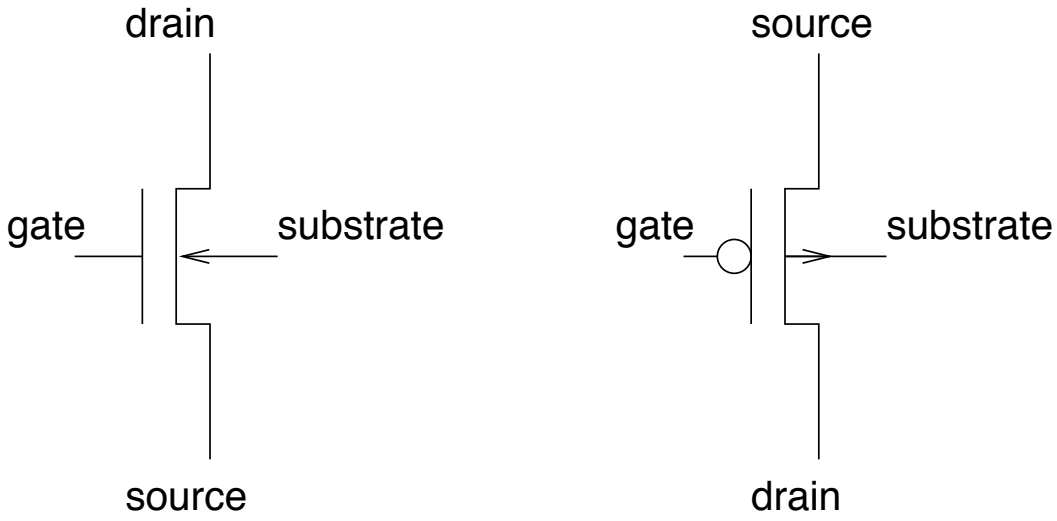
Because we can design systems abstractly and because we can scale, the things we can learn from designing a Tinychip are relevant to research and product development in current and future CMOS VLSI technologies.

We can use:

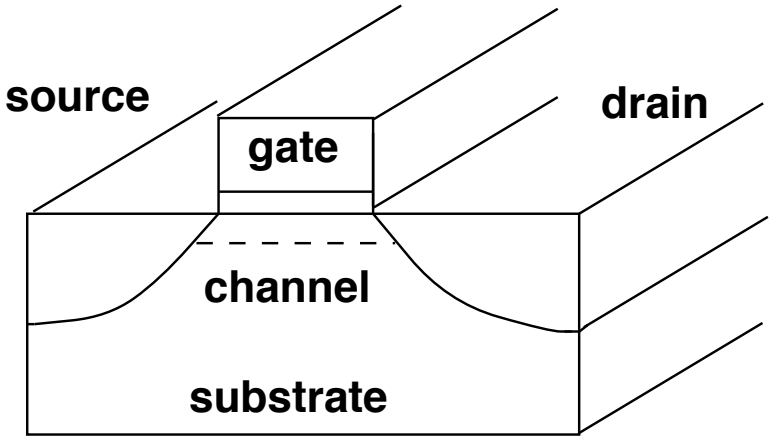
- The same design tools.
- The same hierarchical design style.
- The same circuits.
- And we confront most of the same problems.

The MOSFET

Our basic building block.

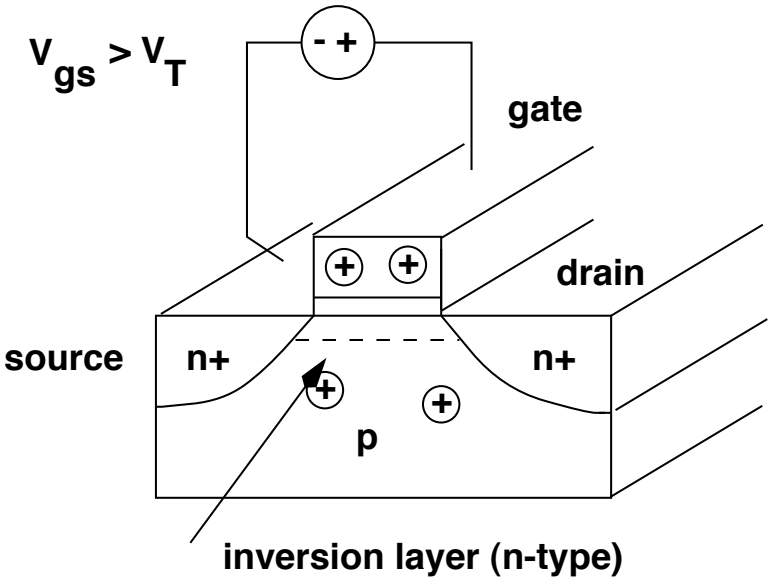


The basic building block in CMOS is the MOSFET, of which we use two kinds: n-channel and p-channel.



How does it work?

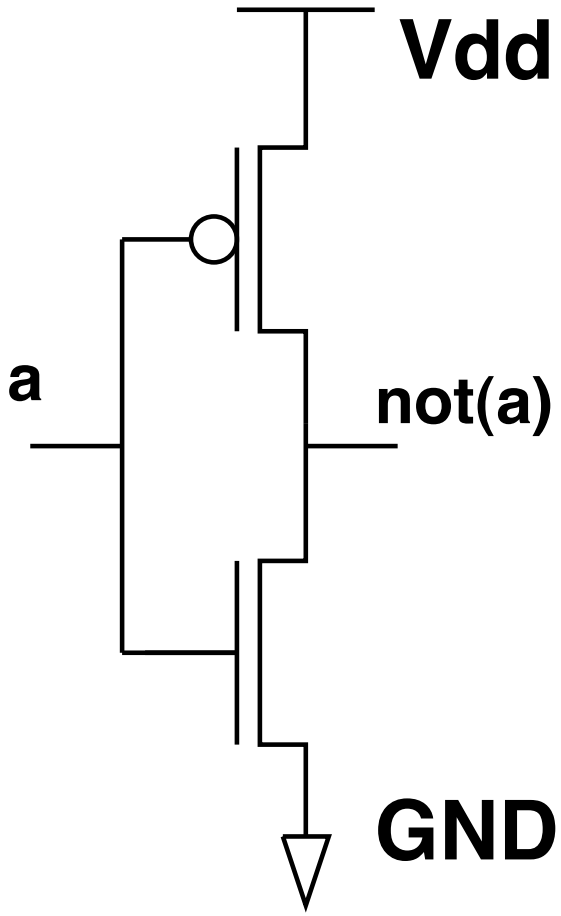
The two types of transistors are used in a *complementary* way. The pFET passes high voltages and the nFET passes low voltages (because of the types of free carriers in the channel).



When enough voltage is applied at the gate terminal, the p-type diffusion between the source and drain is robbed of its holes (they are pushed down) and turns into an n-type semiconductor—the reverse biased p-n junctions at the source and drain are bypassed and the MOSFET turns on.

An inverter

Since the nFET conducts *low* voltages and is turned on by a *high* voltage (and vice versa for the pFET), CMOS logic is inverting (or antimonotonic). We can build things like inverters:



What CS/EE 181 is not about

Some of the topics we will not cover in the class include

- Industrial design tools—Synopsys, Cadence
- Standard HDLs—VHDL, Verilog
- FPGAs, ASICs

Next Time

One-hour crash course on CMOS circuit design.

- Transistors and their properties.
- Noise margins and principles of systematic logic design.
- `magic` layout.
- switch-level simulation with `irsim`.

Other important things to remember:

- First two homework assignments out; get started early...
- TA office hours: TBA.

Change of schedule: Next time (Wednesday) Magic tutorial in the lab!