# Packaging

Jake Beal, April 11, 19100

Today: we'll talk about why we should care about packaging in the first place, followed by what distinguishes good packaging, and then explore some of the design space.

# Why care about packaging?

(And what does “packaging” mean?)

Keep the silicon safe?

Give us a couple communications pins?

Place to put the heatsink?

We don’t really like to think about packing in the CS world. Really, it’s a MechE kinda thing. Put it in a box with wires, maybe make it hard to break, and you’re good.

## Packaging is everything but the silicon!

- Big Beige Box
- Inter-Component connections (e.g. bus)
- Inter-Chip connections (e.g. FTC)
- Intra-Chip connections (e.g. BEOL)

## But why is good packaging necessary?

The Big Beige box isn't a very interesting topic: Apple's just about got that all dealt with. The other stuff is where the meat of the talk is.

# Answer: We have to move information.

- Slow communication = death
- Communication density $\alpha$ complexity
- Bad connections make noise
- Most failures are broken wires

## We need good wires. How many?

If you've got a system that fits on a single chip, all you have to worry about is the beige box. That's no problem. But the bigger (and more interesting) your system is, the more pieces it's going to have, and the more likely those pieces are going to have to share information.

Lousy connections have high impedence, make noise, slow down the rate at which you can transmit data. It's the wires, stupid!

# Rent's Rule:

$$N_p = K_p N_g^{\beta}$$

Warning: Rent's rule isn't real!

Np is the number of pins, Kp and B are proportionality constants, Ng is the number of gates. Discovered back in 1960. Just like Moore's law, it's entirely empirical! Rent found Kp=2.5, B=0.6. Modern stuff goes anywhere from that to 0.45 for processors and 0.12 for SRAM. In fact, it's worse than that since most pins are PWR, GND.

Even though it's not real, it still makes a good guideline and points towards the real problem: moving information around. We need lots of wires, and they must be good.

# What are good wires?

- Short
- High conductivity
- Low noise
- Connected reliably

*speed* $\alpha$ *(# wires) / (wire length)*

First is obvious: distance = latency. High conductivity and a range of other materials properties regulate how fast information propagates in the wires. Low noise is about crosstalk, mostly. We can solve that with all of the PWR/GND wires acting as EMF shielding. Reliable connections is about how good your connections are when you plug things together. Bad connections cause noise, failure, power-expenditure.

## More issues:

- Heat dissapation
- Planar manufacturing technology
- Debugging and testing
- Scale changes - too many wires!
- Manufacturing and assembly
- Palmtop to AHP and beyond

This is a big grab-bag of everything, mostly about the fact that these systems have to live in an imperfect 3D world. Heat dissapation is obvious: even though microchannels are great, we've got to have space to add them. Remember physics.

Also, we've still only got 2D manufacturing in silicon. Even our extra 1/2D is just wires. We'll come back to that. Debugging and testing are also highly dependent on having sufficient access to the chip.

Next, there's the problems that come when we leave the VLSI world. We can bring lots of wires to the edge of a chip, but can we connect them? Worse, the chips have to be plugged in by humans or machines. How do you make them get the wires right? Finally, there's commercial reasons we want machines to be small & hardy.

# Design Approaches

- VLSI (BEOL)
- Perfect Pins (FTC)
- 3D (Neo-Wafer)

The three papers we read each typify nicely a general approach to the packaging problem. As we go through them, watch how the complexity gets pushed around.

VLSI: Get rid of the packaging problem by fitting more on one chip. Great as far as it goes, giving us nice solutions for small-sized problems.

Perfect Pins: Make communication high quality and fast with huge numbers of good pins. FTC is exemplary manufacturing technology too.

3D: shorten all the wires. Make it small. Make it happy. We'll demolish it in time.

VLSI = intrachip. Perfect pins = interchip. 3D = intercomponent.

# VLSI: Violating Rent's Rule

Key idea: Reduce communication problems by getting more on one chip.

- VLSI is compact, efficient.
- No manufacturing or scale problems.

Problem: We've still got to move data, and our die is 2D!

VLSI is great stuff. It makes most of our packaging problems just go away. But VLSI itself is a packaging problem. We solve the 2D thing by adding layers, giving 2-1/2D, and now the key issue is how many layers of wires we need, and how small the wire.

# How many layers?

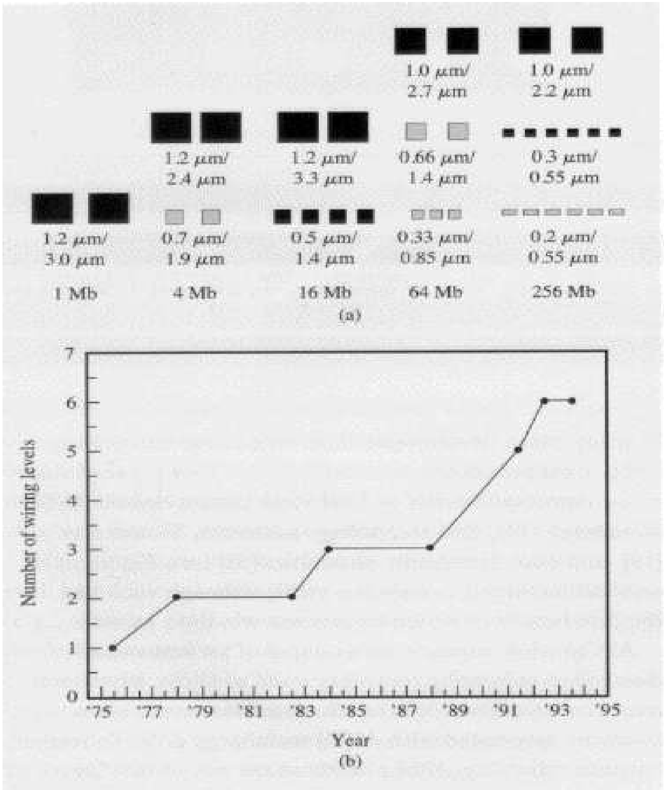


Figure 3

(a) Schematic of wiring levels of 1Mb–256Mb IBM DRAM products. Levels depicted in black represent Al(Cu)-based wiring; those depicted in grey represent W-based wiring. Values of wire

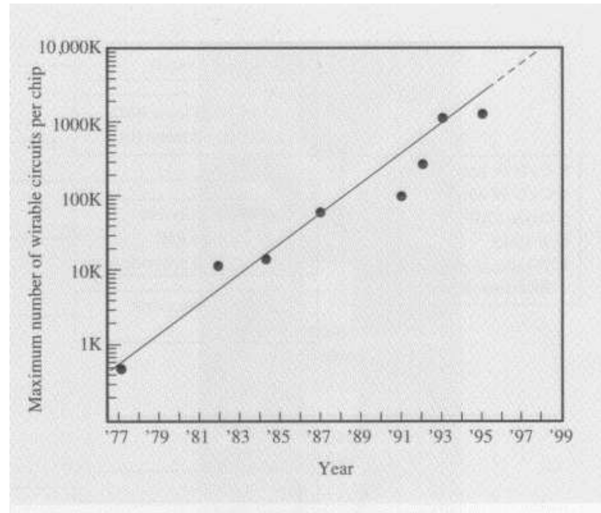


Figure 4

Maximum number of wirable logic circuits per chip vs. year of introduction into IBM manufacturing.

Notice that the maximum wireables corresponds with how many layers there are. Part of this is just due to the better technology with time, but it's my impression that this is really because the limiting factor in silicon area utilization is the wiring. Can I get confirmation from TK & co? BEOL also interfaces with external package.

So what're the factors involved in getting more, tighter layers?

# Getting High Wirability

- Number of Layers

  (High interconnect: Processor)

- Contacted Metal Pitch

  (High density arrays: DRAM)

- Constraints: reliability, speed, cost

The more complex your circuit topology, the more layers you'll need. The more elements you want to pack close together, the higher your pitch (wires per inch) has to be. Of course you always need a good helping of both, but those are the boundary cases driving the problem.

Processors are a good example of high interconnect: locally they aren't too horrible, but they have lots of long wires. RAM has a very simple circuit topology, but the devices are very small, and the whole goal is to maximize the number per area.

# Stacking Up Layers

## Planarity, planarity, planarity!

- CVD
- CMP
- Damascene

Planarity gives us lots of wonderful things. The flatter our layers are, the better we can stack them: match the vias, have more precision interconnects. Error builds up with every successive level. Planarity means we've got it dealt with.

Chemical Vapor Depositing means that we can lay down layers nicely.

Chemical Mechanical Polishing means that we can smooth down a layer after we deposit it.

Damascene is a technique where we lay wires into trenches instead of building it on the surface.

# Perfect Pitch

Much, much harder than good planarity

Increasing demand - low power computing means less silicon per device.

Requirements:

1. Good planarity

   Higher error = looser pitch

High pitch is complicated by lots and lots of issues! Not only that, but it's needed more and faster as low-power computing gets serious. Low power gates take up less area by a square factor, but we can only use as many as we can connect!

To get good pitch, the first thing we need is good planarity. As we go up in layers, error gets cumulatively worse. Wire pitch is bounded by the precision with which we can place it.

# Perfect Pitch- Planarity

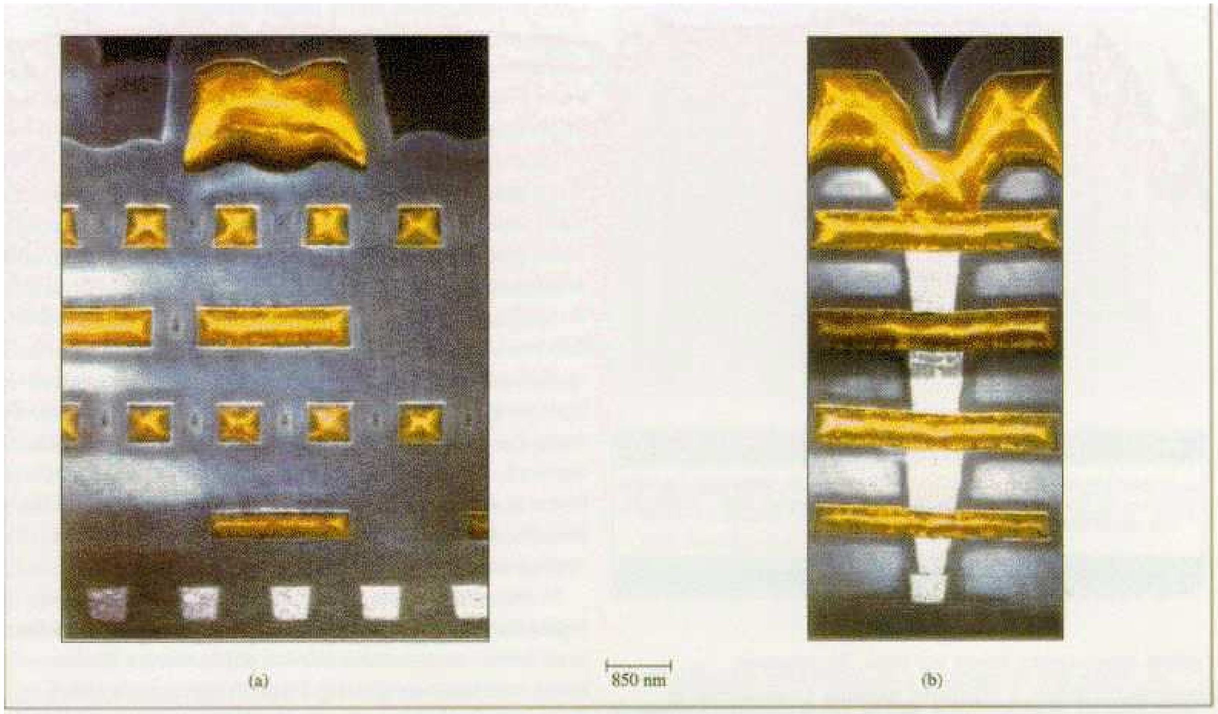


Figure 6

Scanning electron micrograph cross sections of structures fabricated using planar BEOL process: (a) BEOL portion of a 0.5-$\mu$m CMOS logic chip currently in production; note that a local interconnect level and five global interconnect levels are used. (b) Stacked contacts and vias. From [11], reproduced with permission.

Note that the higher levels are less pretty! Particularly important are the vias on the right

# Perfect Pitch

2. High reliability

Electromigration, mechanical stress, manufacturing error

(Remember: failures are broken wires!)

$TiAl_3$/Al(Cu) and Ta/Cu wires - no Ag!

Tungsten (W) via studs

Dual-layer oxide deposition

Ti/Al(Cu) is changing to Ta/Cu. Silver sucks for oxidization AND cost. Dual-layer oxide with CMP between helps prevent shorts.

# Perfect Pitch

3. Resistivity and Capacitance

   Speed, power - wire width

   Crosstalk - wire separation

   Ta/Cu wires

   Low dialectric insulators:

   organic/organosilicate polymers?

TK also mentioned something about beryllium crystalling wires last time.

Insulators reduce capacitance and crosstalk. Looking at dialectric < 3!

# Perfect Pins: Big Bandwidth

Key idea: Scrupulously follow Rent's Rule

- BIG systems HAVE to deal with wiring

NEC SX-3, Tera

Essentially, this is what we traditionally think about as "packaging", and really it just comes down to "if you're gonna do it, damn well do it right."

# Issues:

- VLSI/chip scale change
- Fitting in all the pins
- Connecting the chip

The scale-change problem is a big one here: VLSI can bring a huge number of connections to the edge, but at the chip level the wires are HUGE. Big idea here is to go for 2D solder-bump connection rather than 1D edge connection (more reliable too).

Fitting in all the pins. How FAR do you have to move the pins from the die? Again, 1D/2D comes in here. Also, what's your pin pitch?

Connecting the chip. The chip is going to have to be placed by humans or robots, and neither are very precise. How do you design your chip so that it doesn't get misaligned or pins broken or whatnot?

# The Elegant SX-3

- FTC has dense 2D connections (although not the TAB LSI)
- Solder-bump connections
- MLS provides lots of nearby bus and fewer distant-communication pins.
- MLS chip footprints, LCM modularity

Really, it's just nice, and sums up just about everything you want to do. The high-density pinouts let the chip funnel lots of data around. The connection design allows high-speed, low noise communication. The bus system lets information move smoothly. The chip footprints and LCM design make it hard to screw up putting things together. It's even set up for good cooling. A nice product.

# 3D: Unobtainium

Key idea: Use the third dimension to compact everything and shorten wires.

- VLSI: break out of 2.5D
- Component-level: better busses

{\Barbie  3D IS HARD!!}

Now for the fun stuff: unobtainium in it's 3D form! There's essentially two attacks on this problem: trying to go 3D with your silicon, and trying to pack your silicon efficiently. Neo-Wafer is really trying to do both. Other systems, including some of the big parallel machines we looked at earlier, have got the component level down pretty well. (exa: J-Machine)

# Why 3D is good:

- 2D backplane
- Bandwidth, bandwidth, bandwidth
- Tiny Beige Box footprint
- Short distances

Most of our communication goes over single wires or, at best, a flat strip of wires. Now, if you can have a nice 3D package such that you can have a 3D backplane, your bandwidth just skyrocketed. Same goes for the VLSI level: cross-sectional bandwidth just goes through the roof!

Then you've got the military thing. Advanced Humanics Project - a good 3D architecture could let you have a normal computer in a really teeny box. Good for Palm people too.

And, of course, distance = time. Good 3D layouts let us put things much closer together - remember, all graphs can be embedded in 3D!

# Why 3D is EVIL:

- Heat dissapation
- Debugging/Testing
- Mindbending design
- Planar processes!

Remember your physics. Heat is dissapated by area and generated by volume.

Debugging is really the big one. You need good access to a chip, and the third dimension is how we generally do that. If we lived in a 4D world, this wouldn't be a problem.

As anybody who's played with VRML or Descent knows, humans are much better at dealing with 2.5D than 3D.

Finally, our manufacturing is still limited to 2D, so there's a limit on how much 3D we can build in the first place.

# Neo-Wafer: a good try!

Neo-Wafer actually seems workable, partly because it's not all that ambitious.

- Really just another backplane bus
- High conductivity, low power

Neo-Wafer is pretty cool, actually. I looked up their web-page and found some updates, but not much useful info. Maybe because it's DoD?

Really, though, it's all about building a standard backplane bus. The cool thing here is that they realized that to connect their different components, they didn't actually need to put them on perf-boards. Testing can be normal.

As to the heat problem, which many people brought up in questions, yes, it is high conductivity, and that's going to solve a lot of the problem. The real answer, though, is that it's just not supposed to consume appreciable power in the first place. So, why dissapate when you can just not generate? DoD influence too.

They don't say they've demoed, which I suspect is due to backplane-building probs.

# Key Ideas:

- You've got to have lots of good wires.
- speed $\alpha$ (# wires) / (wire length)
- Hire some MechE talent.
- 3D is hard: don't get ambitious