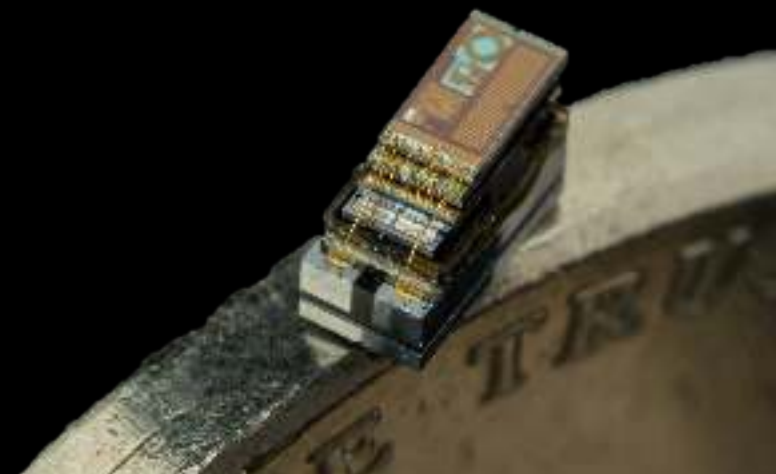
Lessons From Five Years of Making Michigan Micro Motes (M^3)

Pat Pannuto, Yoonmyung Lee, ZhiYoong Foo, Gyouho Kim,
David Blaauw, and Prabal Dutta

University of Michigan

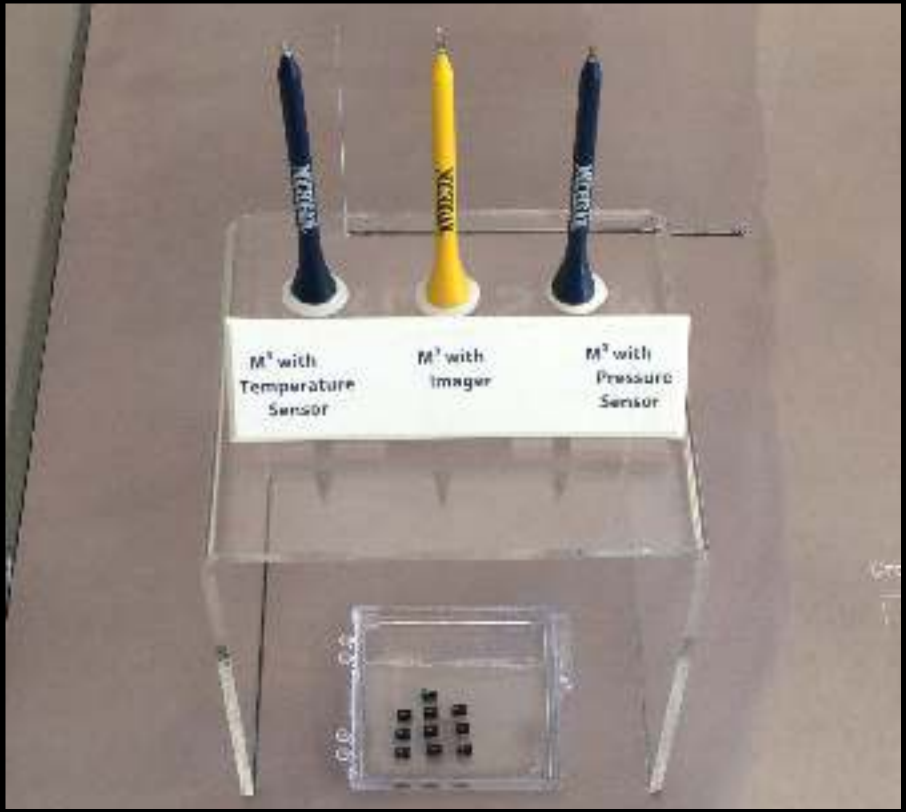
6th Workshop on Architectural Research Prototyping (WARP '15)

June 14, Portland, Oregon, USA



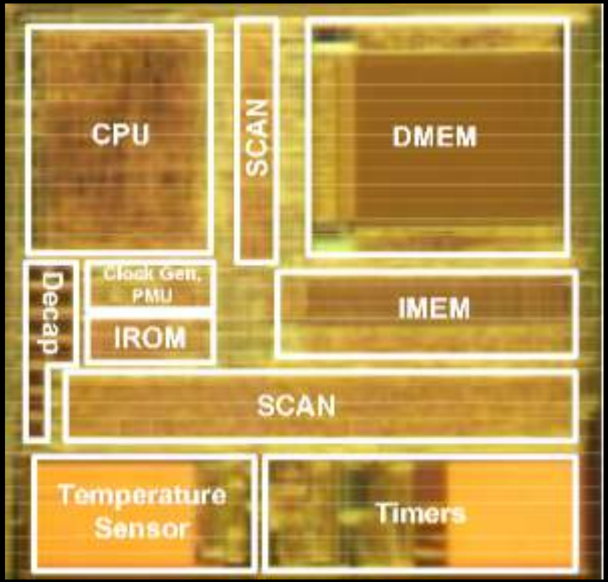
Behind the scenes on what was required to realize an ecosystem of mm-scale computers

On display now at the computer history museum



2008: The Phoenix Processor

- World's lowest power computer
  - Set records for **volume** and **energy**
  - And could be used by at best the grad students that built it




Mingoo Seok


Scott Hanson


Yu-Shiang Lin


ZhiYoong Foo


Daeyeon Kim


Yoonmyung Lee


Nurrachman Liu

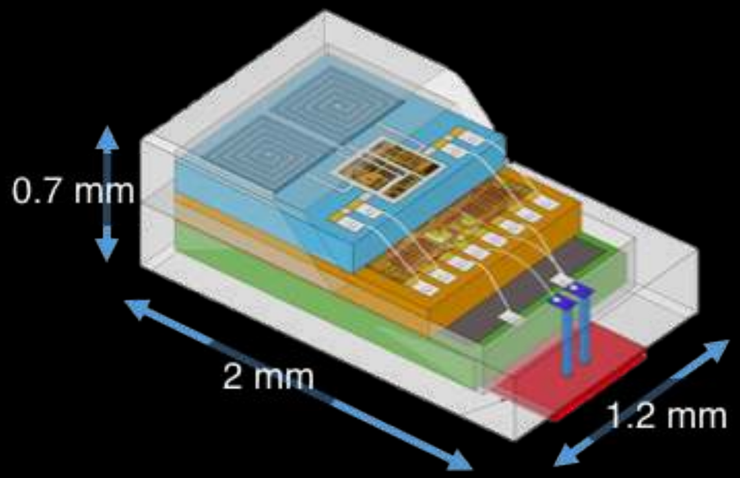

Dennis Sylvester

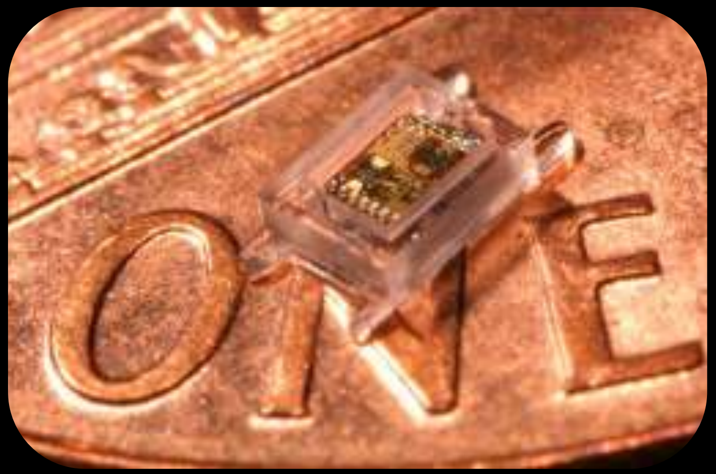

David Blaauw

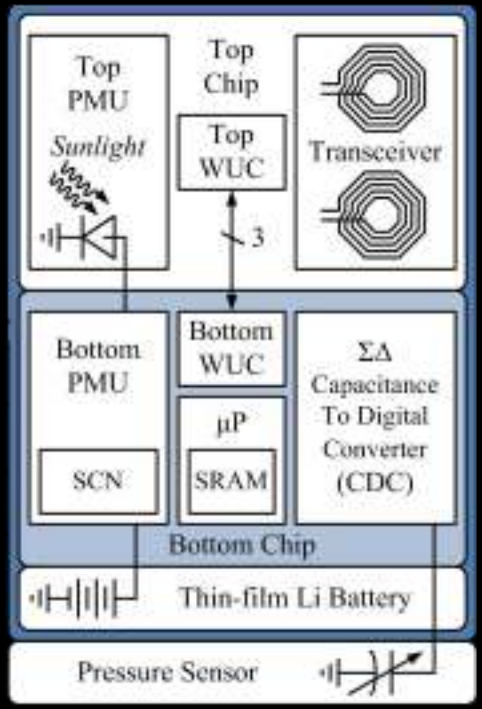
The Phoenix Processor: A 30pW Platform for Sensor Applications, M. Seok, VLSI '08

From temperature sensor to pressure sensor: **3 Years.**

- Motivating application for mm-scale sensor: Intraocular pressure
  - Critical diagnostic tool for glaucoma







A Cubic-Millimeter Energy-Autonomous Wireless Intraocular Pressure Monitor, G. Chen, ISSCC '11

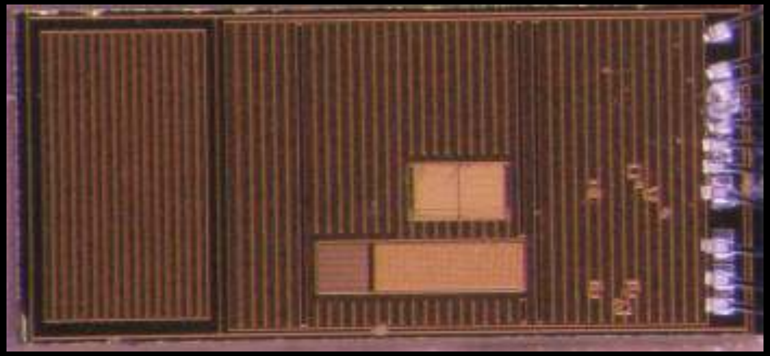
What's wrong with this picture:
"How to build a wireless temperature sensor?"

cm-scale

- Step 1: Buy a low-power temperature sensor

mm-scale

- Step 1: Design a low-power temperature sensor



~10 pW standby, < 1 μW active

What's wrong with this picture:
"How to build a wireless temperature sensor?"

cm-scale

- Step 1: Buy a low-power temperature sensor
- Step 2: Buy a low-power microcontroller

mm-scale

- Step 1: Design a mm-scale temperature sensor
- Step 2: Design a mm-scale CPU

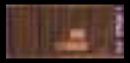
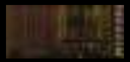
~10 pW standby, < 1 μW active

What’s wrong with this picture:
“How to build a wireless temperature sensor?”

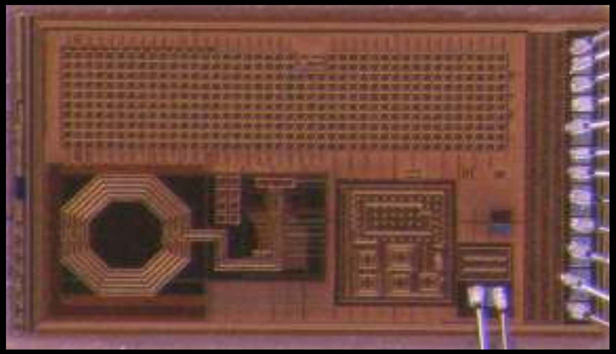
cm-scale


- Step 1: Buy a low-power temperature sensor
- Step 2: Buy a low-power microcontroller
- Step 3: Add low-power communications

mm-scale

- Step 1: Design a mm-scale temperature sensor
- Step 2: Design a mm-scale CPU
- Step 3: Design a mm-scale radio



~10 pW standby, ~5 μW active

What's wrong with this picture:

"How to build a wireless temperature sensor?"

cm-scale

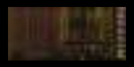
- Step 1: Buy a low-power temperature sensor

- Step 2: Buy a low-power microcontroller

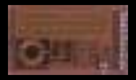
- Step 3: Add low-power communications
- Step 4: Add a power supply

mm-scale

- Step 1: Design a mm-scale temperature sensor



- Step 2: Design a mm-scale CPU



- Step 3: Design a mm-scale radio
- Step 4: Add power supply and storage

Energy Harvesting

Energy Storage

What’s wrong with this picture:
“How to build a wireless temperature sensor?”

cm-scale

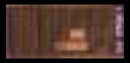
- Step 1: Buy a low-power temperature sensor

- Step 2: Buy a low-power microcontroller

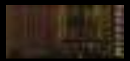
- Step 3: Add low-power communications

- Step 4: Add a power supply
- Step 5: Integrate the **chips**

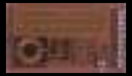
mm-scale



- Step 1: Design a mm-scale temperature sensor

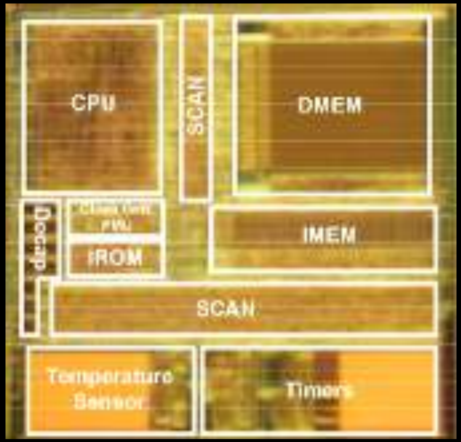


- Step 2: Design a mm-scale CPU



- Step 3: Design a mm-scale radio

- Step 4: Add power supply and storage
- Step 5: Integrate the **modules**



What’s wrong with this picture:

“How to build a wireless temperature sensor?”

cm-scale – Modular System

- Step 1: Buy a low-power temperature sensor

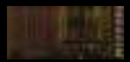
- Step 2: Buy a low-power microcontroller

- Step 3: Add low-power communications

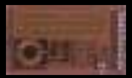
- Step 4: Add a power supply
- Step 5: Integrate the **chips**

mm-scale – Monolithic System

- Step 1: Design a mm-scale temperature sensor

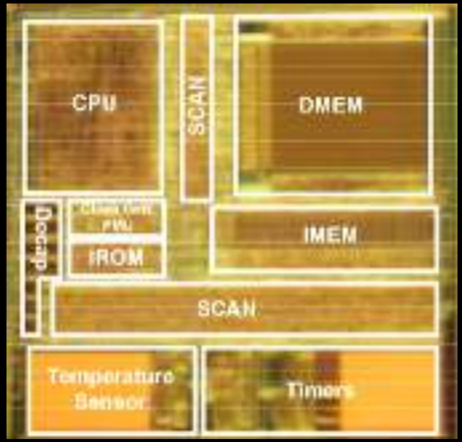


- Step 2: Design a mm-scale CPU



- Step 3: Design a mm-scale radio

- Step 4: Add power supply and storage
- Step 5: Integrate the **modules**



What’s wrong with this picture:
“How to build a wireless temperature sensor?”

cm-scale – Modular System

- Integrate components using well-defined, standard interfaces
  - **SPI**
  - **UART**
  - **I²C**

mm-scale – Monolithic System

- Integrate components using Verilog
  - **temp_sense t1 (**
    ...
    .sample_valid_out (to_proc_t1_sample_valid),
    .sample_out (to_proc_t1_sample),
    );
  - **processor p1 (**
    ...
    .temp_valid_in (to_proc_t1_sample_valid),
    .temp_in (to_proc_t1_sample),

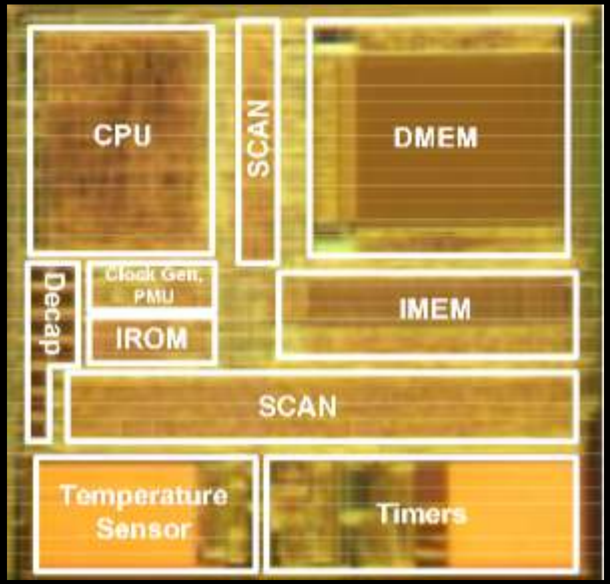
What’s different?

What's wrong with this picture:
"How to build a wireless temperature sensor?"

cm-scale – Modular System

- Integrate components using well-defined, standard interfaces
  - ~~SPI~~
  - ~~UART~~
  - I^2C

mm-scale – Monolithic System



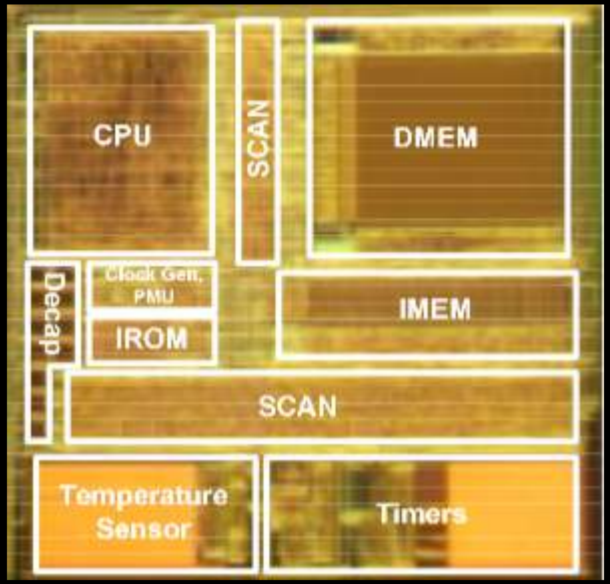
Previously unseen constraints on volume

What's wrong with this picture:
"How to build a wireless temperature sensor?"

<u>cm-scale</u> – Modular System

- Integrate components using well-defined, standard interfaces
  - ~~SPI~~
  - ~~UART~~
  - ~~I^2C~~

<u>mm-scale</u> – Monolithic System



Previously unseen constraints on volume and energy

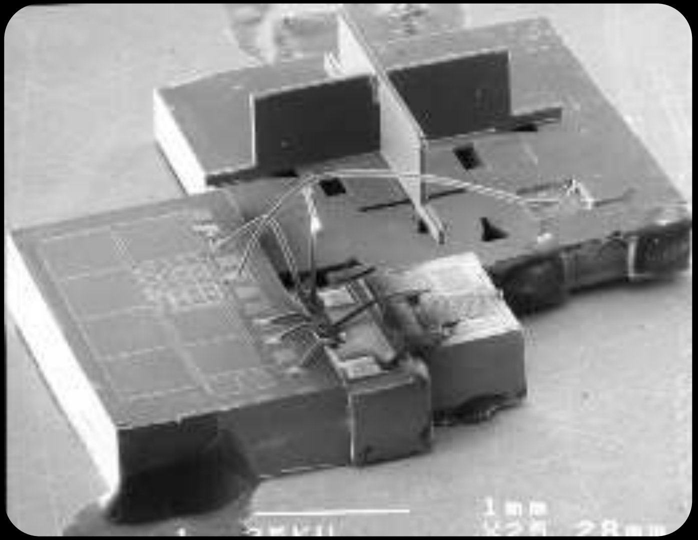
What's wrong with this picture:

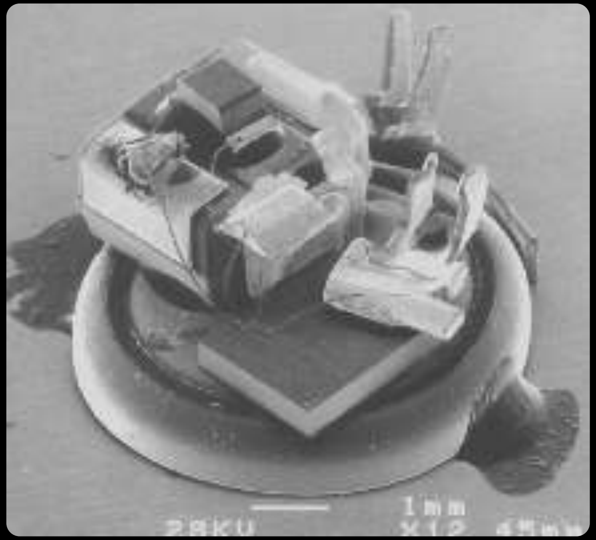
"How to build a wireless temperature sensor?"

cm-scale – Modular System

- Integrate components using well-defined, standard interfaces
  - ~~SPI~~
  - ~~UART~~
  - ~~I^2C~~

mm-scale – "Modular" Systems





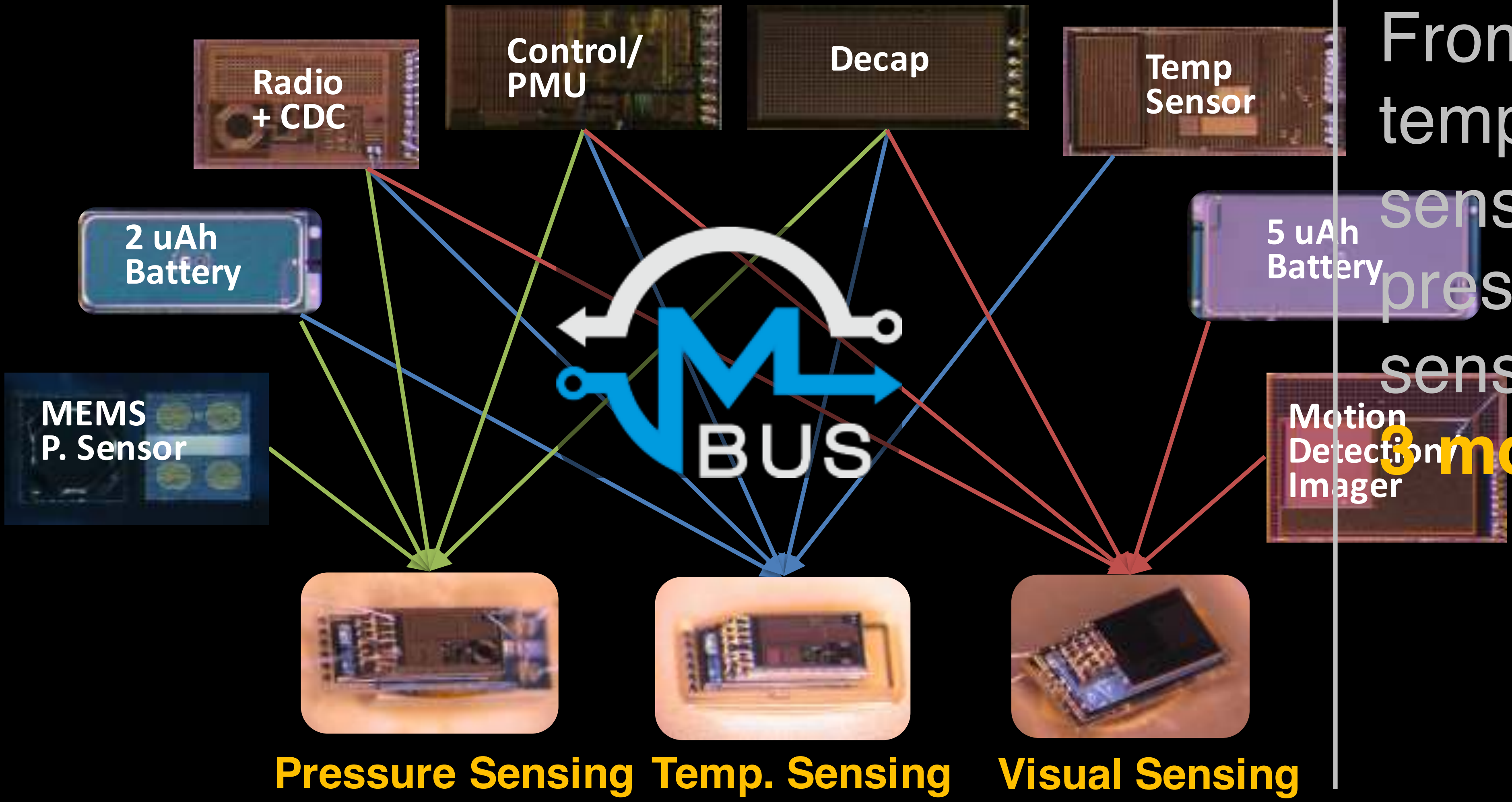
Ad-hoc composition results in poor use of volumetric space

MBus, the missing interconnect that enables the mm-scale computing class

- 22.6 pJ / bit / chip, < 10 pW standby / chip
- Single-ended (push-pull) logic
- Low, fixed wire count (4)
- Multi-master
- **Power-aware**

- Implemented in over a dozen (and growing) mm-scale **chips**
  - CPU, Radio
  - Flash Memory
  - Temperature, Pressure, Imager
- To make half a dozen (and growing) mm-scale **systems**

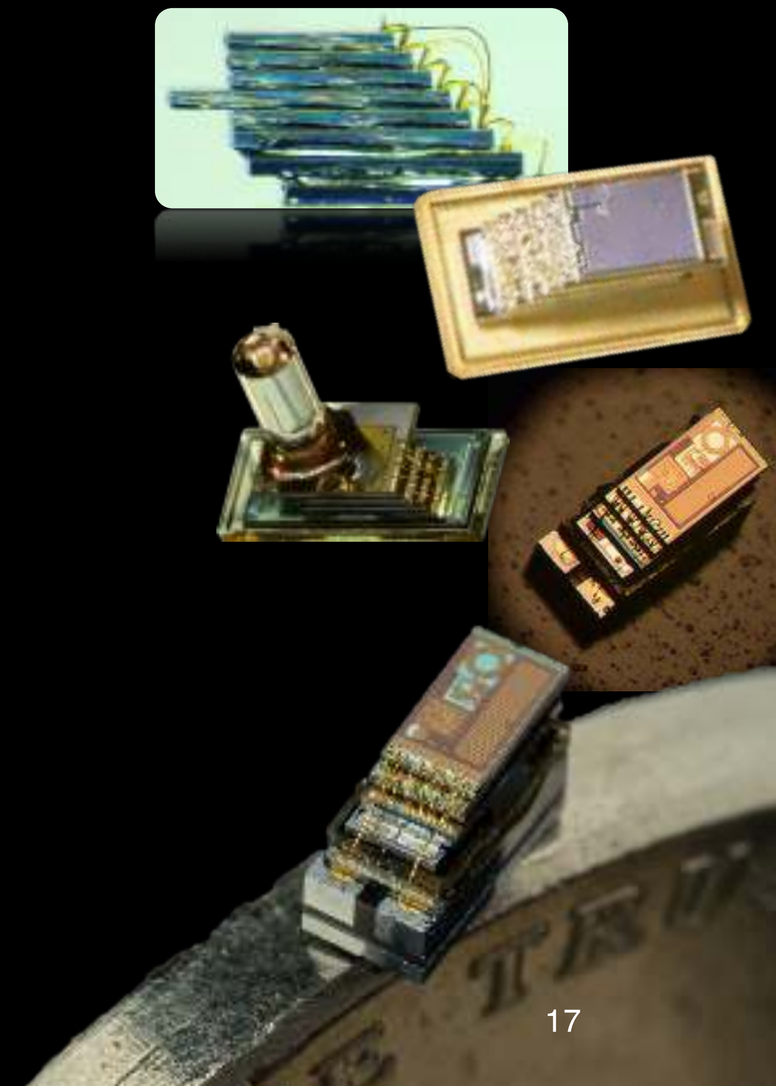
MBus enabled modular components, key to the rapid development cycle we now enjoy



From temperature sensor to pressure sensor: **3 months.**

Learn more about MBus at ISCA

- Session 10B
  - MBus: An Ultra-Low Power Interconnect Bus for Next Generation Nanopower Systems
- Wednesday, 13:50 in Oregon 204

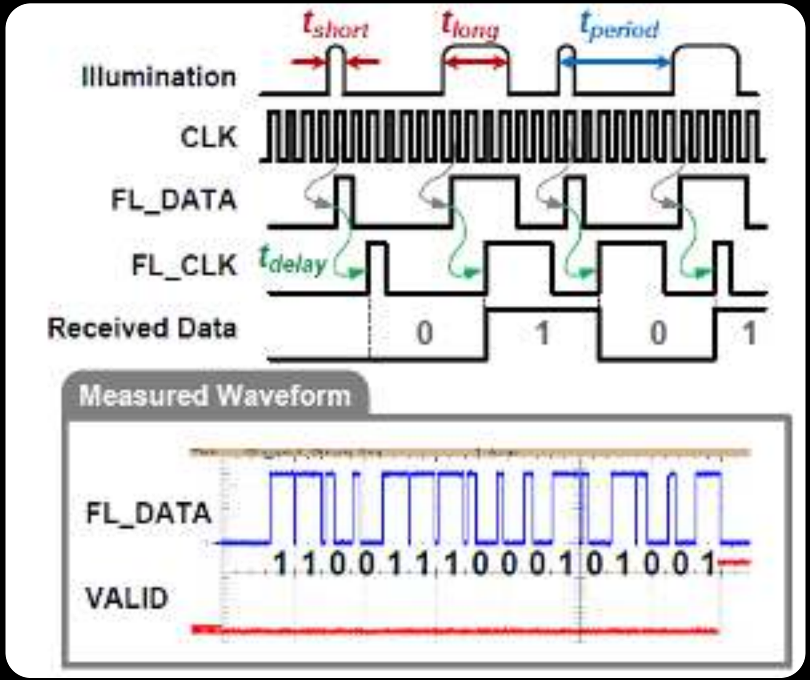


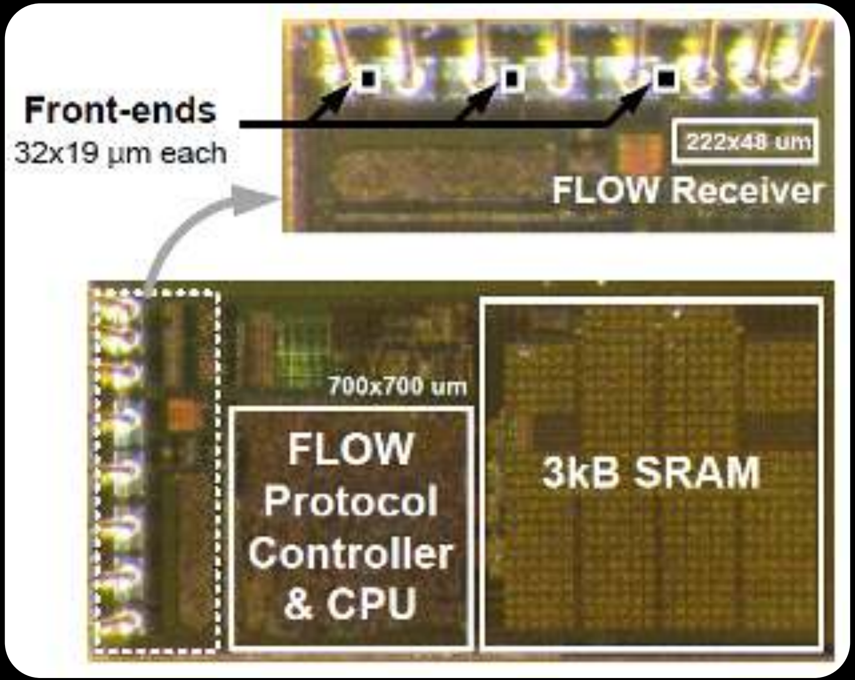
Digging into what a mm-scale system looks like and why that really matters

- Flashy pictures from a good camera can be misleading.
- In practice, several nodes fit on a period

How do you bootstrap (program) a node that’s too small to attach wires to?

- Visible light communications



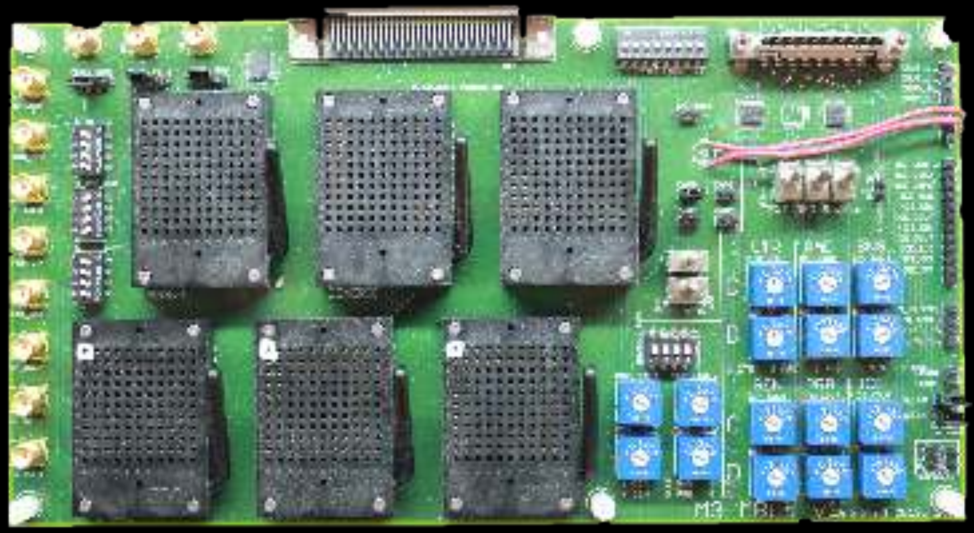


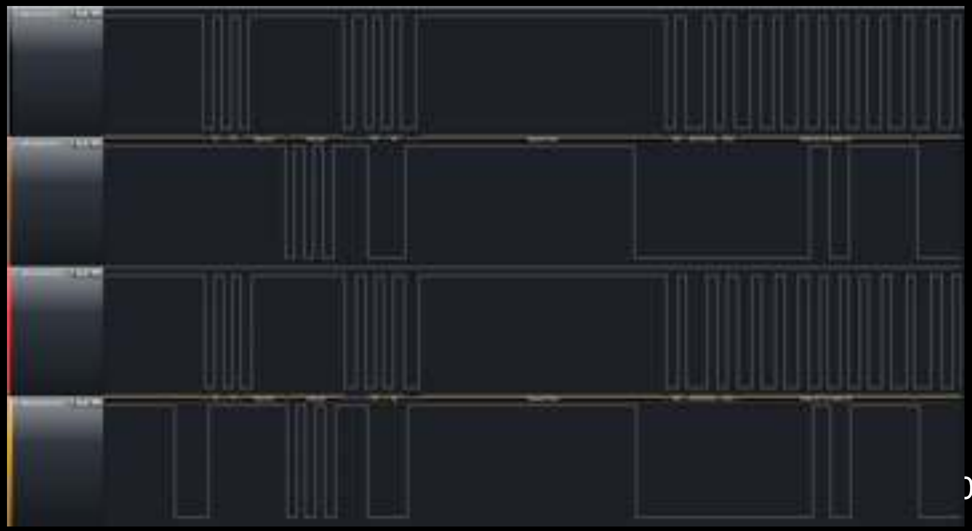
Gyouho Kim, Yoonmyung Lee, Suyoung Bang, Inhee Lee, Yejonng Kim, Dennis Sylvester, David Blaauw, “A 695 pW Standby Power Optical Wake-up Receiver for Wireless Sensor Nodes,” IEEE Custom Integrated Circuits Conference (CICC), September 2012

Open Q: How do you debug a node that's too small to attach wires to?

- Partial answers
  - Breakout system
  - Visible light debugging
    - Inject arbitrary messages
  - Power trace debugging
    - Each state has a power signature
  - Bus snooping
    - Plug-in for Saleae logic analyzer

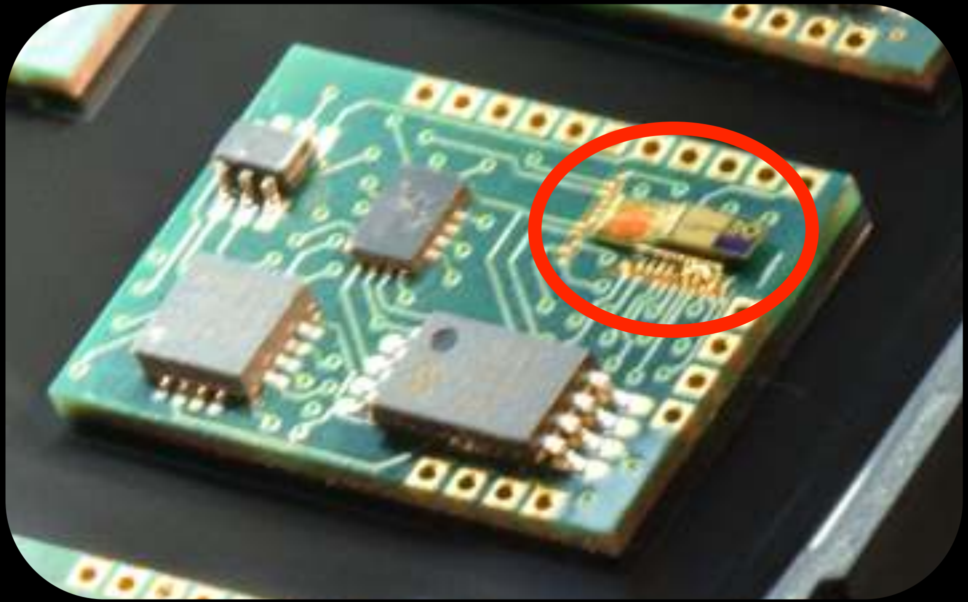
- **Big Open Q: Postmortem**

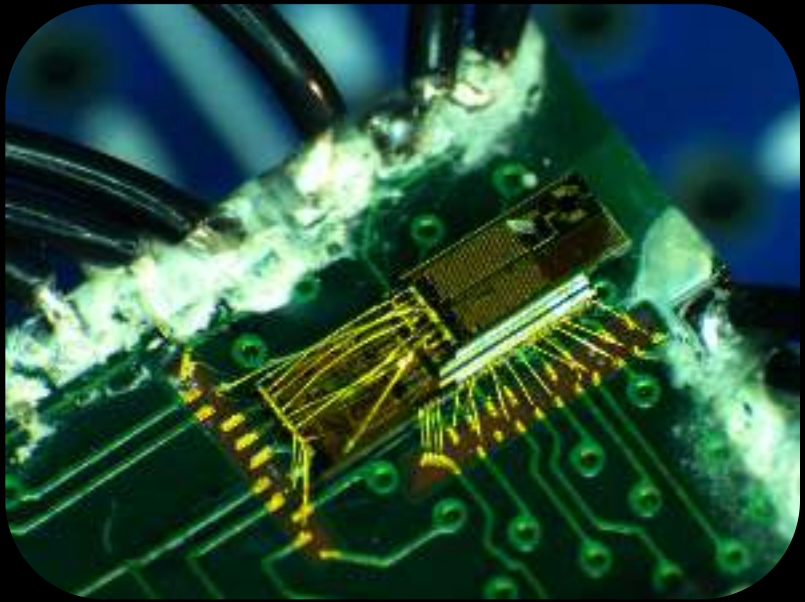




[Recent efforts] How do you solder a node that's too small to attach wires to?

- Integrating mm-scale chips with COTS chips



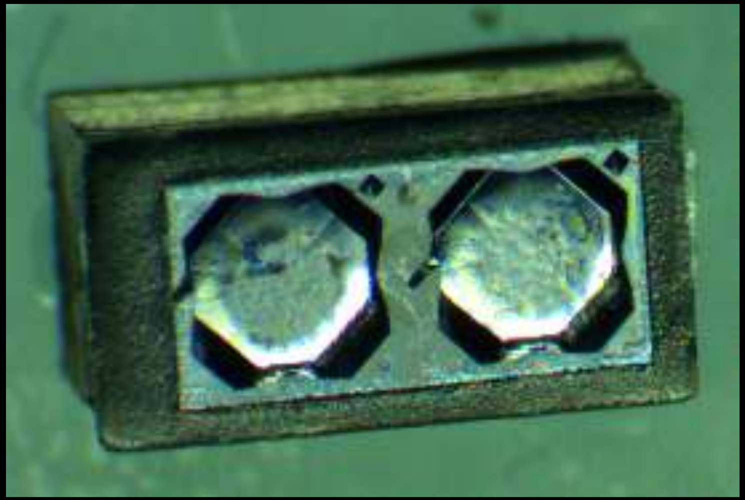


- Recent issues: Metals for **both** soldering and wirebonding

(to 50 µm pads)

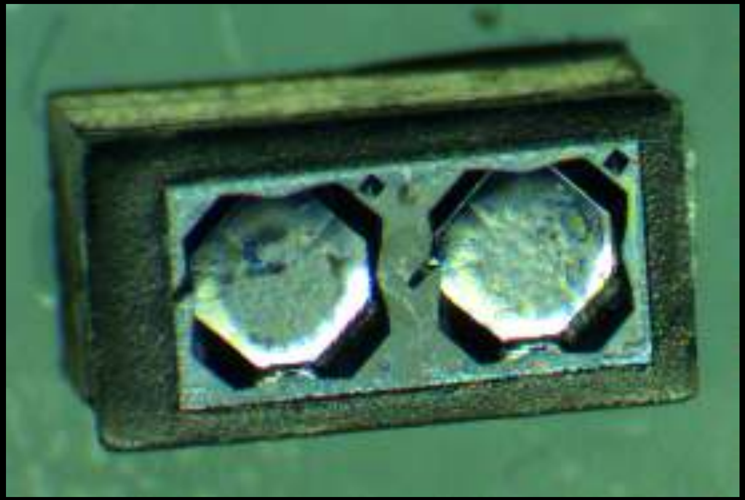
Re-discovering packaging problems from the MEMS community

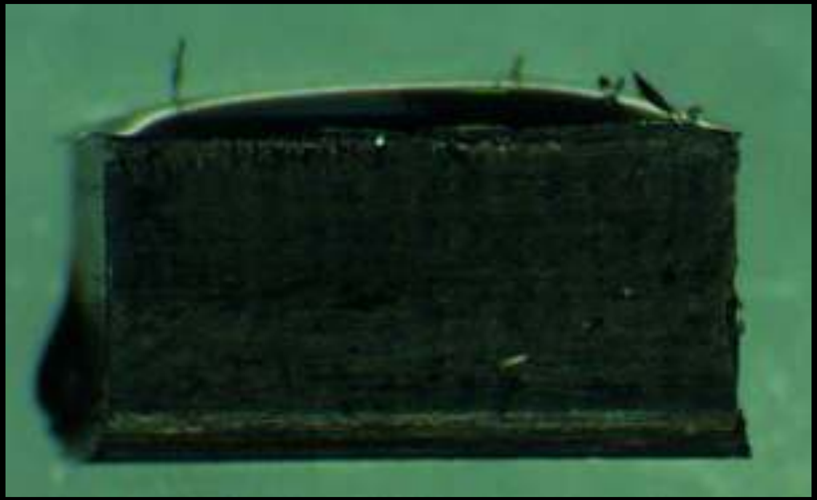
- Bio-compatible pressure sensor
  - Can seal electronics in bio-compatible epoxy
  - *How to expose the pressure sensor?*



And adding some new packaging problems of our own

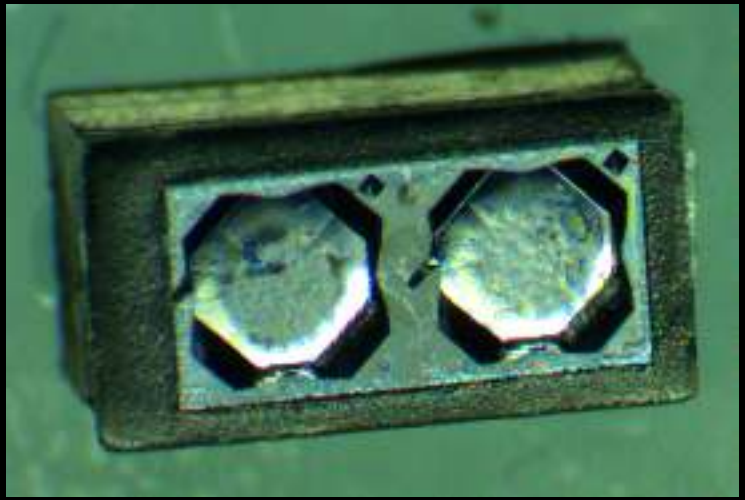
- Bio-compatible pressure sensor
  - Can seal electronics in bio-compatible epoxy
  - How to expose the pressure sensor?
  - *Some of the electronics are light sensitive*

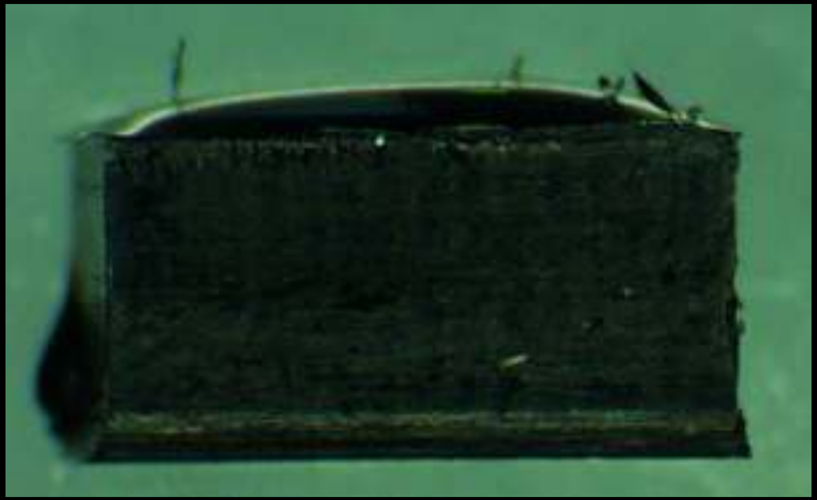


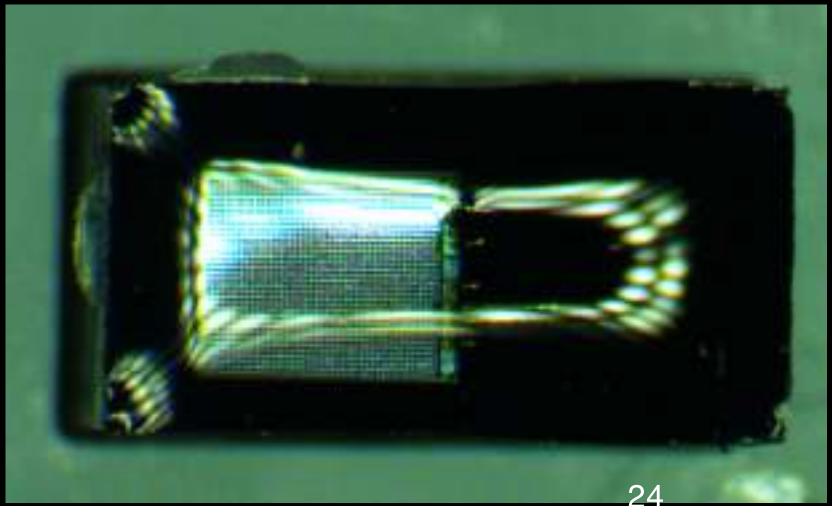


Systems considerations fit into the packaging puzzle too

- Bio-compatible pressure sensor
  - Can seal electronics in bio-compatible epoxy
  - How to expose the pressure sensor?
  - Some of the electronics are light sensitive
  - *The node relies on solar (or IR) harvesting to charge [and program…]*

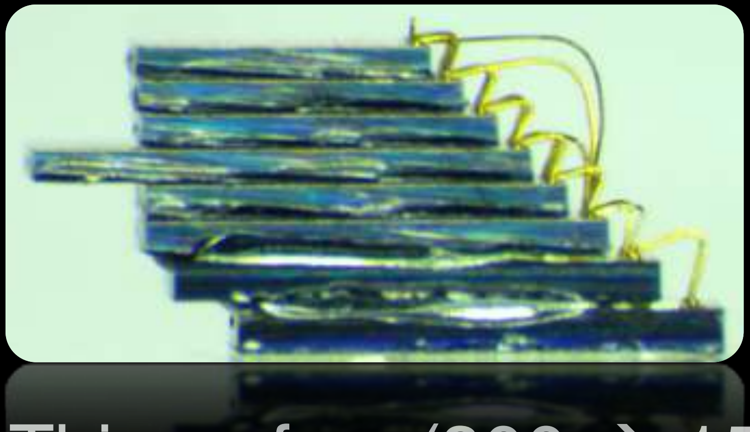




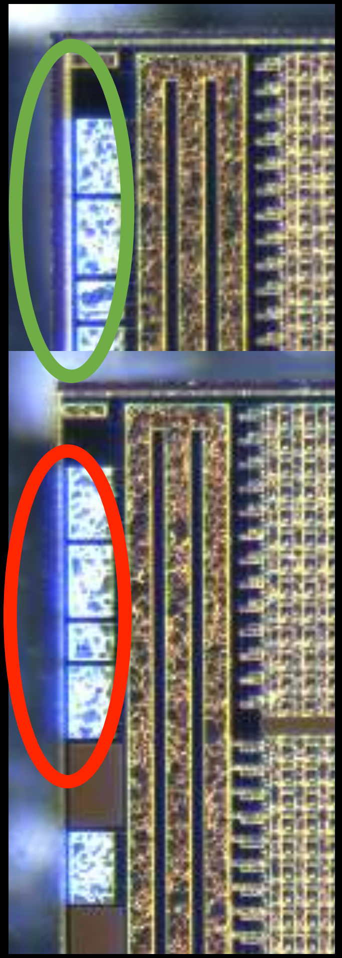


How do you reliably manufacture a mm-scale system?

- Most systems are “stair-step” stack of chips



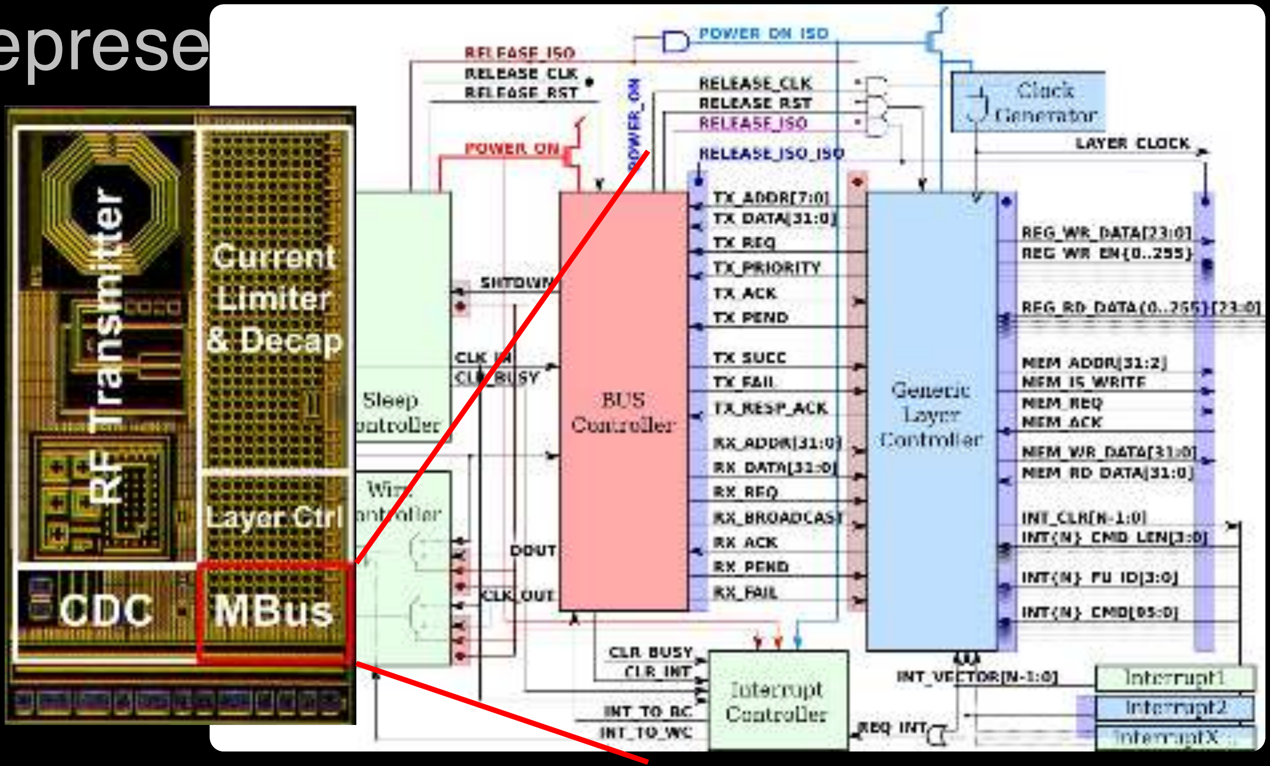
- Thin wafers (300 → 150 microns) [now brittle]
- Then dice on die-attach film
- Then flip onto stack and wirebond



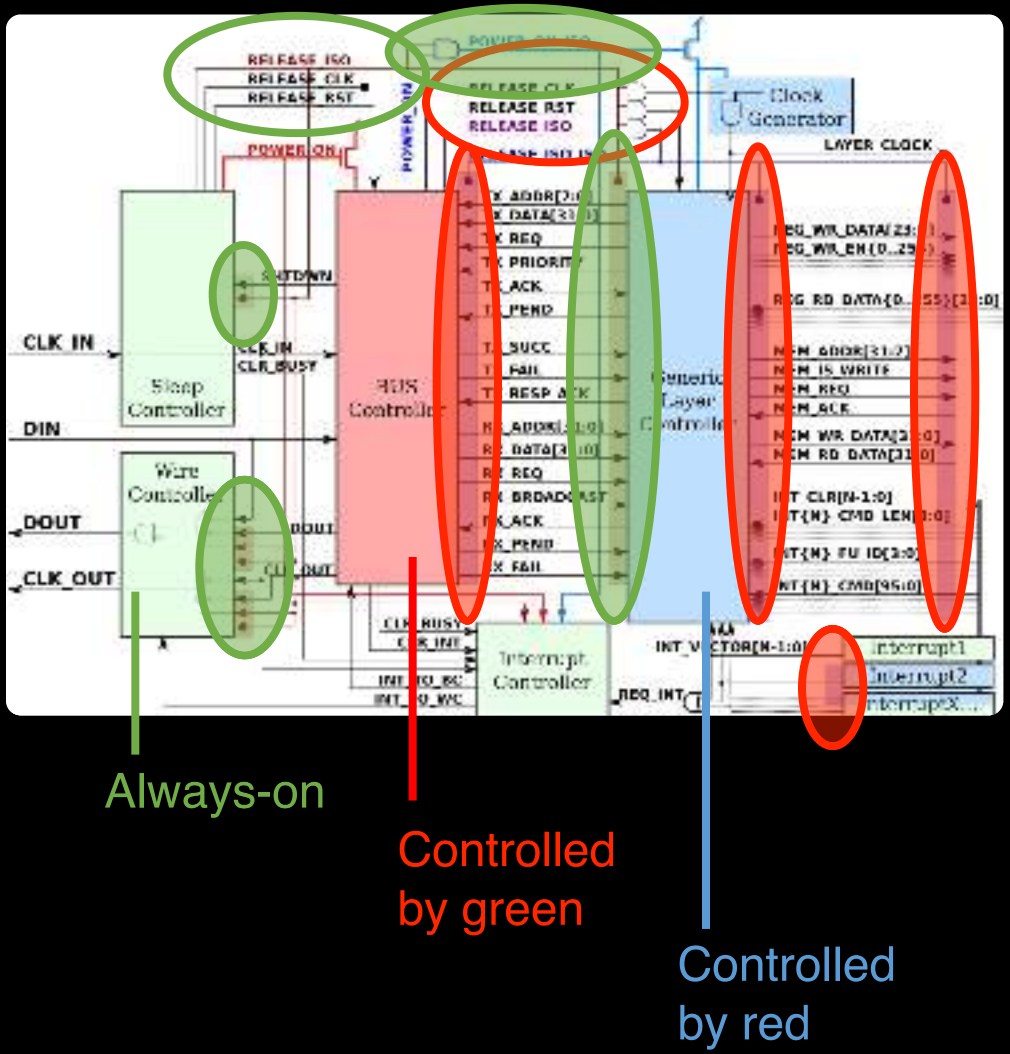
Each step requires some learning…

How do you reliably design a mm-scale system?

- Conceptually simple: Turn off what you aren’t using
- In practice, lots of diagrams that look like this:
- Which represe



Clock-gating != Power-gating. Need to eliminate static leakage as well

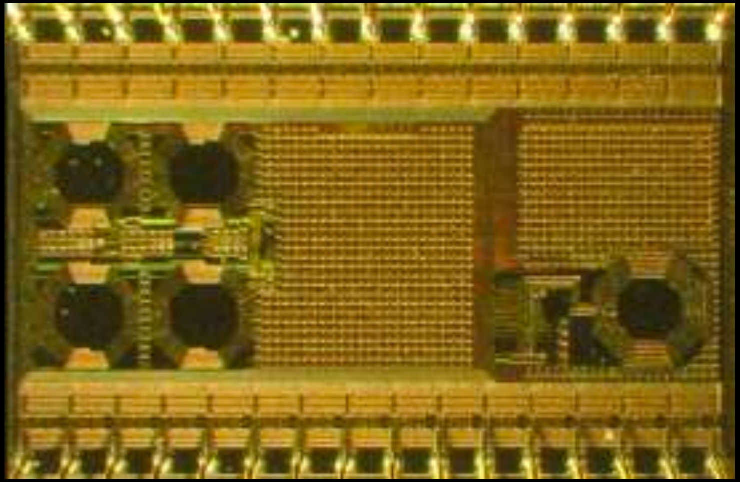


- What is this diagram actually showing?
  - Three hierarchical power domains
  - Power control signals between domains
  - Isolation gates between domains
- Absence of tool support is biggest challenge
  - Isolation correctness validated by hand
  - Effects of isolation simulated “by hand”
    - Often not at all

How to bring a new chip into M^3 ecosystem?

R1: Dedicated test chip

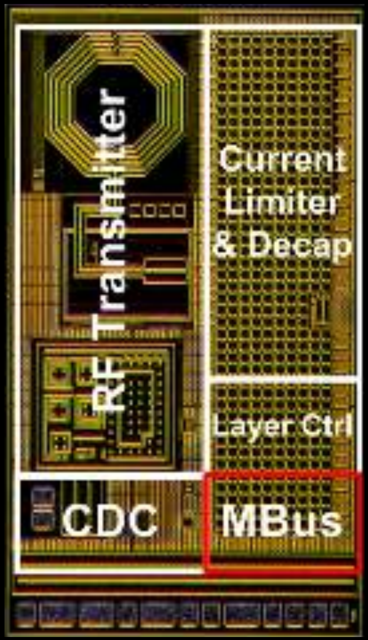
- Lots of I/O, scan, etc



Mohammad Hassan Ghaed, Skylar Skrzyniarz, David Blaauw, Dennis Sylvester, "A 1.6nJ/bit, 19.9µA Peak Current Fully Integrated 2.5mm2 Inductive Transceiver for Volume-Constraned Microsystems," IEEE Custom Integrated Circuits Conference (CICC), September 2014

R2: Integrate

- Add MBus frontend
- Remove most debug pins, expose via registers



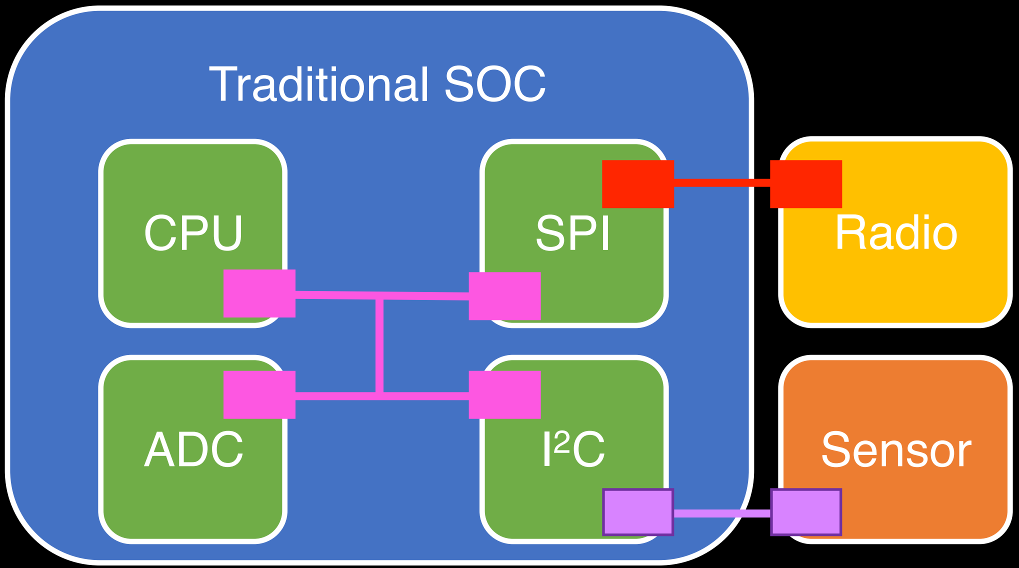
R3+: Bugs!

THERE IS A NATURAL TENSION BETWEEN BUILDING WORKING SYSTEMS AND WORKING CHIPS

Need to avoid designing **“modular” chips** that only work in **one system**

MPQ: A protocol standard for register and memory access brings SOC efficiency to multi-chip systems

- Architectural shift: Remove CPU from steady-state operation
  - Pre-programmed commands:
    - ADC: “Every 128 samples, send DMA to radio”
    - RF: “On DMA CH1 event, send buffer as payload of PKT_TEMPLATE_1”
  - Atmel’s “sleepwalking”, TI’s µDMA controller



SOCs:

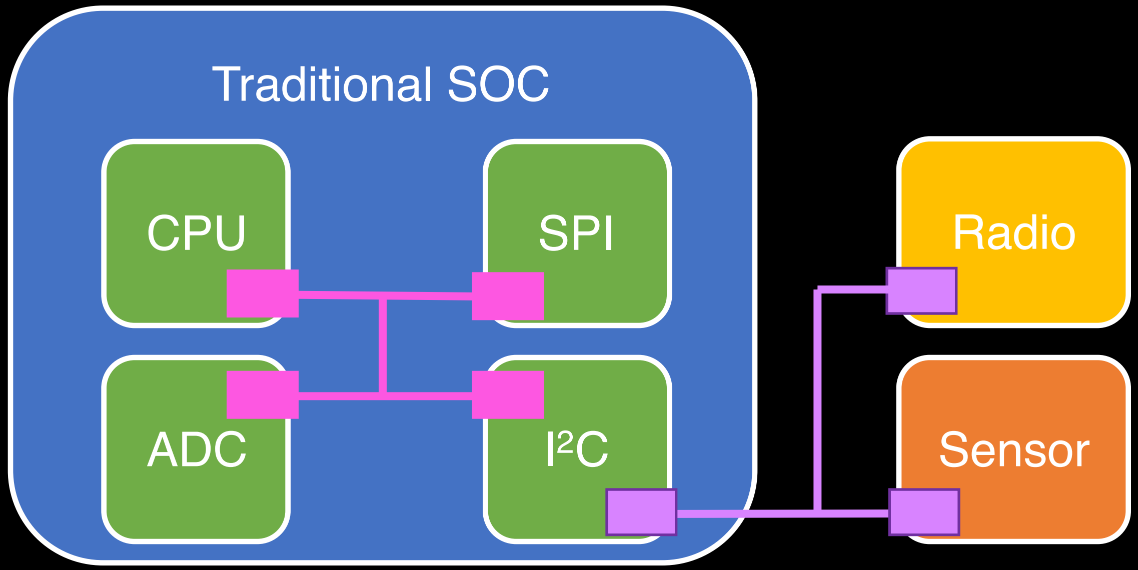
Efficiency demands pushing designs back to a monolithic architecture

But the extensible parts aren’t efficient!

- CPU can’t program sensor to send packet

MPQ: A protocol standard for register and memory access brings SOC efficiency to multi-chip systems

- Architectural shift: Remove CPU from steady-state operation
  - Pre-programmed commands:
    - ADC: “Every 128 samples, send DMA to radio”
    - RF: “On DMA CH1 event, send buffer as payload of PKT_TEMPLATE_1”
  - Atmel’s “sleepwalking”, TI’s µDMA controller



The issue isn’t the physical layer

The clean break of a new bus is an opportunity to standardize basic constructs

- MPQ – “For those who mind their P’s & Q’s on MBus”
  - Does not specify device behavior, command interface, etc
  - Only specifies
    - What a “register” looks like
    - What “memory” looks like
    - And how to read and write registers and memory
- *This is enough to enable powerful system synthesis constructs*

And aligns with how systems are already designed today

MBus and MPQ are available today: mbus.io

- “MBus Compatible”
  - Logo certification
- Specification and Verilog available (free forever)

MBus + MPQ: Standard interface for mm-scale components

It takes a village...


Mingoo Seok


Scott Hanson


Yu-Shiang Lin


ZhiYoong Foo

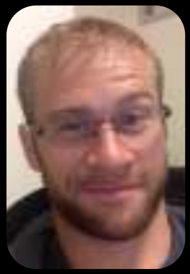

Daeyeon Kim


Yoonmyung Lee


Nurrachman Liu


Dennis Sylvester


Suyoung Bang


Ryan Rogel


Hyeongseok Kim


David Blaauw


Gregory Chen


Razi-ul Haque


Michael Wieckowski


Yejoong Kim


Gyouho Kim


Prabal Dutta


Inhee Lee


Pat Pannuto


Jonathan Brown


David Wentzloff


Elnaz Ansari


Ye-Sheng Kuo


Benjamin Kempke


Wanyeong Jung


Yen-Po Chen


Seokhyeon Jeong


Kuo-Ken Huang


Ronald Dreslinski


Dongmin Yoon


Mohammad
Hassan Ghaed


Myungjoon Choi