

*Note: This presentation uses animation in several places. If you are giving this presentation, I highly encourage you to use the powerpoint version, available at <http://patpannuto.com/talks>*



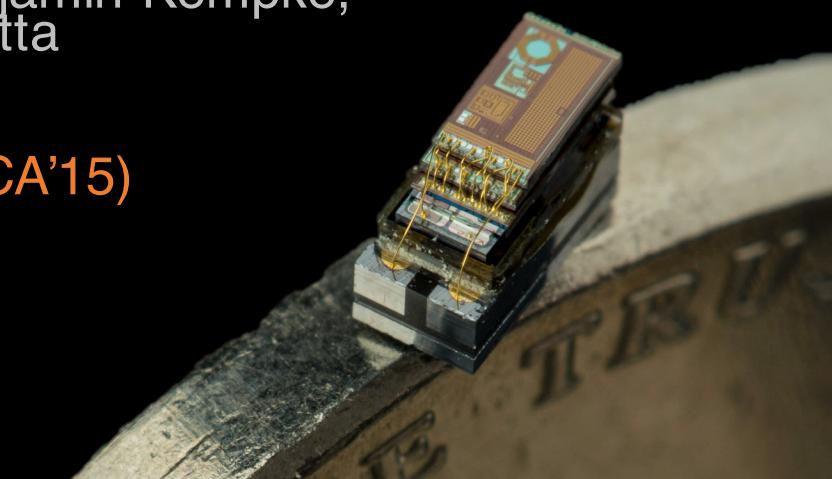
# MBUS

## AN ULTRA-LOW POWER INTERCONNECT FOR NEXT GENERATION NANOPower SYSTEMS

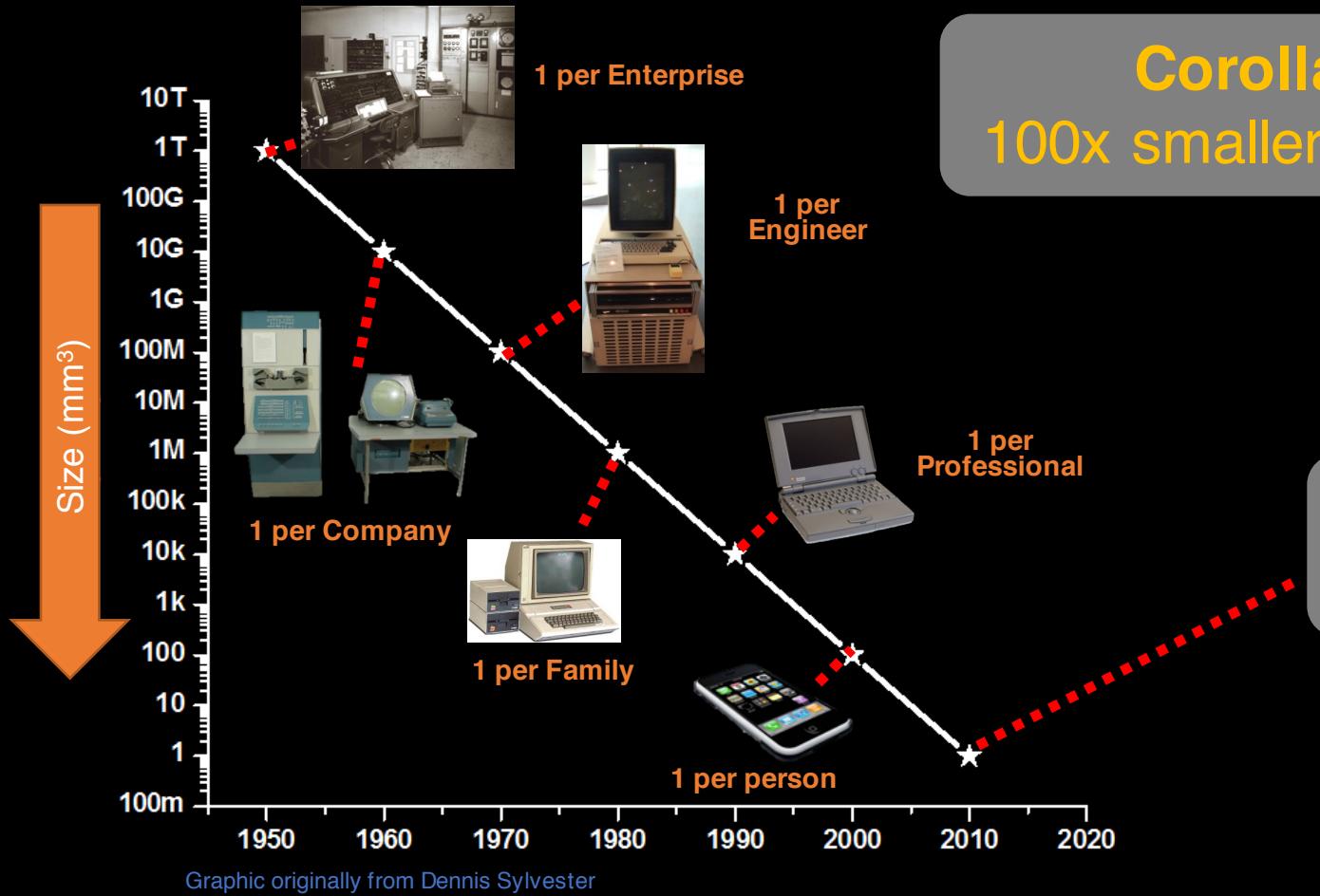
Pat Pannuto, Yoonmung Lee, Ye-Sheng Kuo, ZhiYoong Foo, Benjamin Kempke,  
Gyouho Kim, Ronald G. Dreslinski, David Blaauw, and Prabal Dutta

University of Michigan

The 42<sup>nd</sup> International Symposium on Computer Architecture (ISCA'15)  
June 13-17, Portland, Oregon, USA



# Bell's Law: A new computing class every decade



**Corollary:**  
100x smaller / decade

BY GORDON BELL

## BELL'S LAW FOR THE BIRTH AND DEATH OF COMPUTER CLASSES

*A theory of the computer's evolution.*

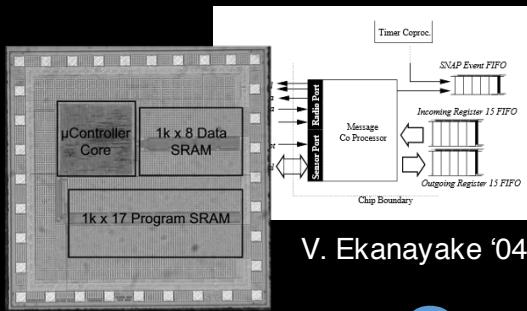
**"Smart Dust"** should  
arrive by ~2017

In the 1950s, one could fit a computer and by 2010 a  
single microchip cluster. By 2030, these processors will have  
expanded to the size of a building. More importantly, computers are begin-  
ning to "swallow" us. This slide illustrates the computing spectrum illus-  
trating the vast dynamic range of computing power, size, cost, and other  
factors for early 21st century computer classes.

A computer class is a set of computers in a particular price range with  
unique or similar programming environments (such as Linux, OS/360,  
Palm, Symbian, Windows) that support a variety of applications that com-  
municate with people and/or other systems. A new computer class forms  
and approximately doubles each decade, establishing a new industry. A  
class may be the consequence and combination of a new platform with a  
new programming environment, a new network, and new interface with  
people and/or other information processing systems.

# We have a diverse array of mm-scale components

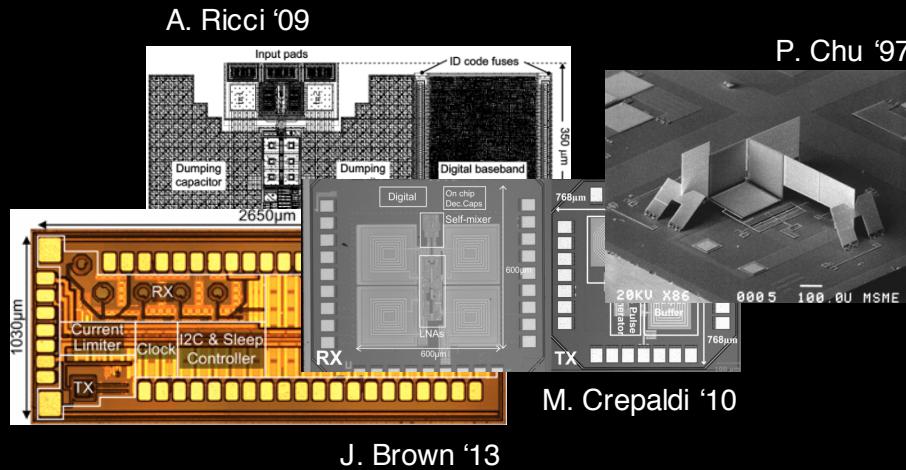
## CPUs



V. Ekanayake '04

B. Warneke '04

## Communication



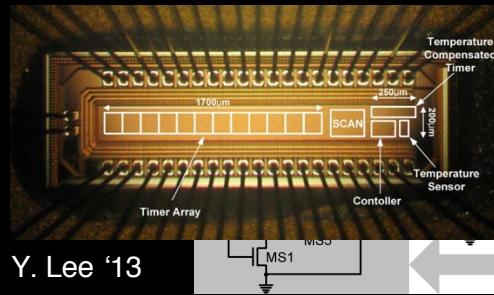
J. Brown '13

M. Crepaldi '10

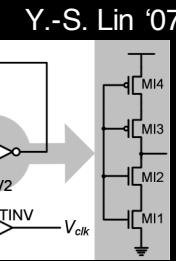
P. Chu '97

A. Ricci '09

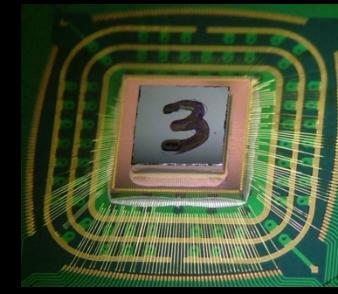
## Timers



Y. Lee '13

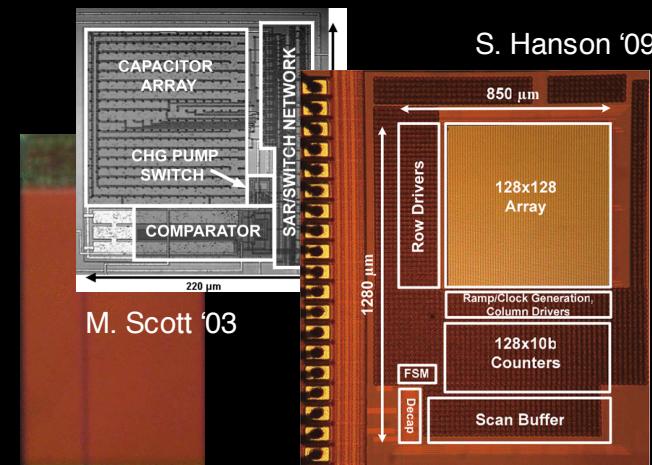


## Power Management



N. Sturcken '13

## ADCs and Sensors



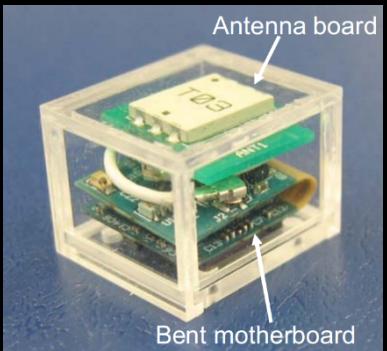
M. Scott '03

Y.-S. Lin '08

S. Hanson '09

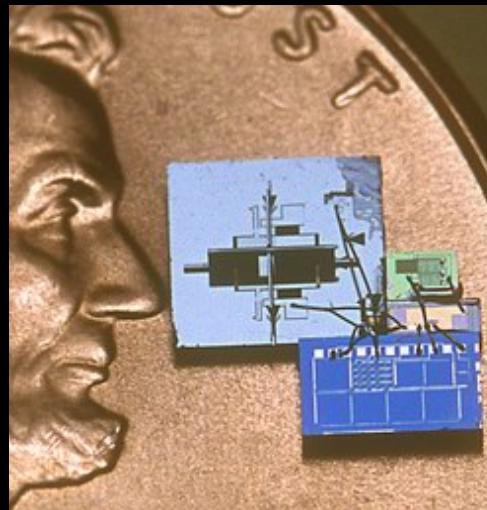
# And a few mm-scale systems...

The 1cc Computer



T. Nakagawa '08

Smart Dust



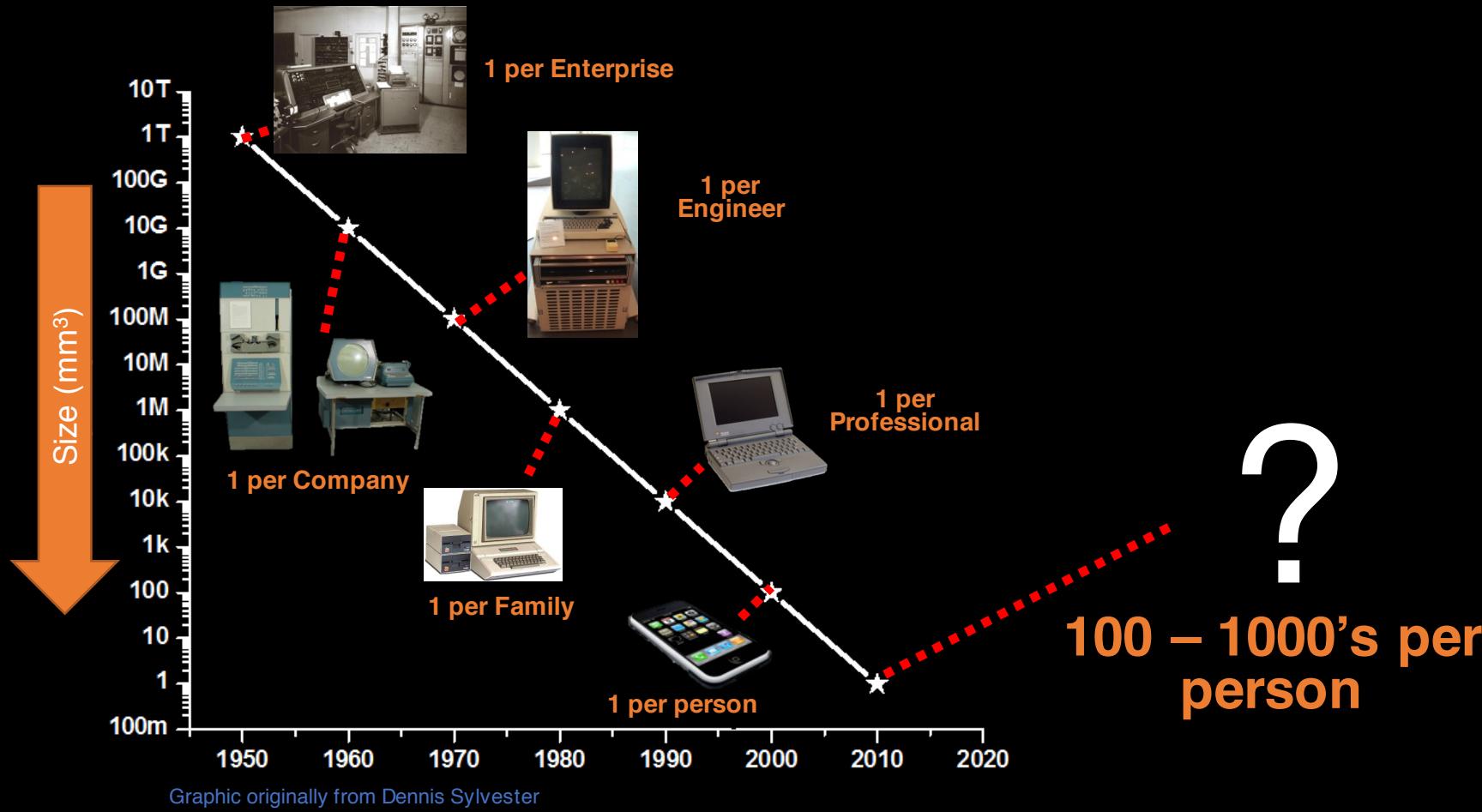
B. Warneke '07

Smart Dew



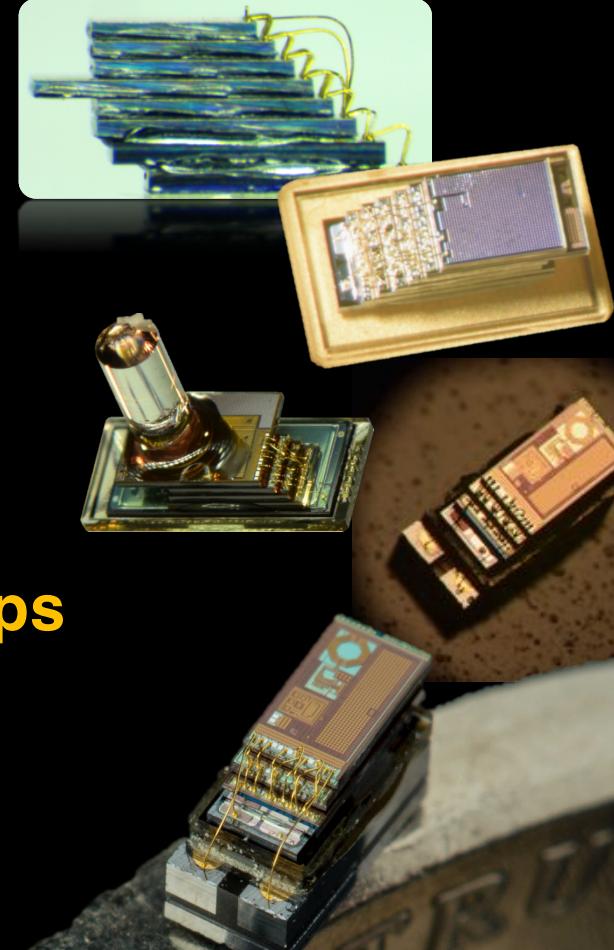
Y. Shapira '08

# But where is the next class of computing?



# MBus is the missing interconnect that enables the mm-scale computing class

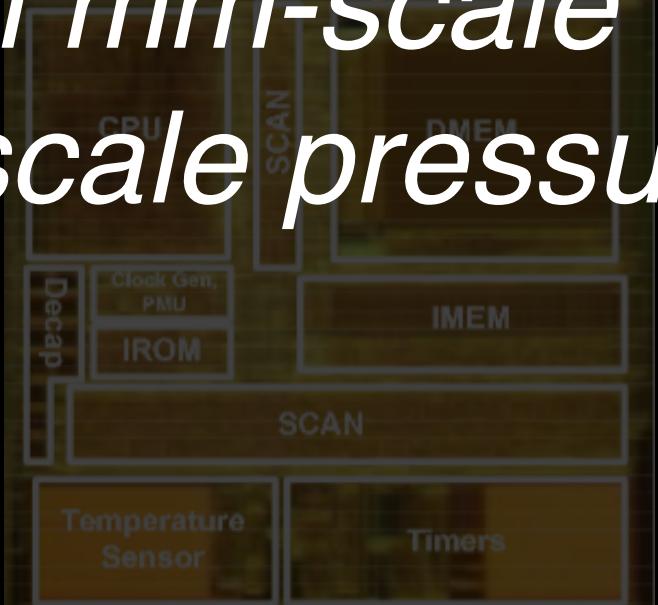
- 22.6 pJ / bit / chip, < 10 pW standby / chip
- Single-ended (push-pull) logic
- Low, fixed wire count (4)
- Multi-master
- **Power-aware**
- Implemented in over a dozen (and growing) mm-scale **chips**
  - CPU, Radio
  - Flash Memory
  - Temperature, Pressure, Imager
- To make half a dozen (and growing) mm-scale **systems**



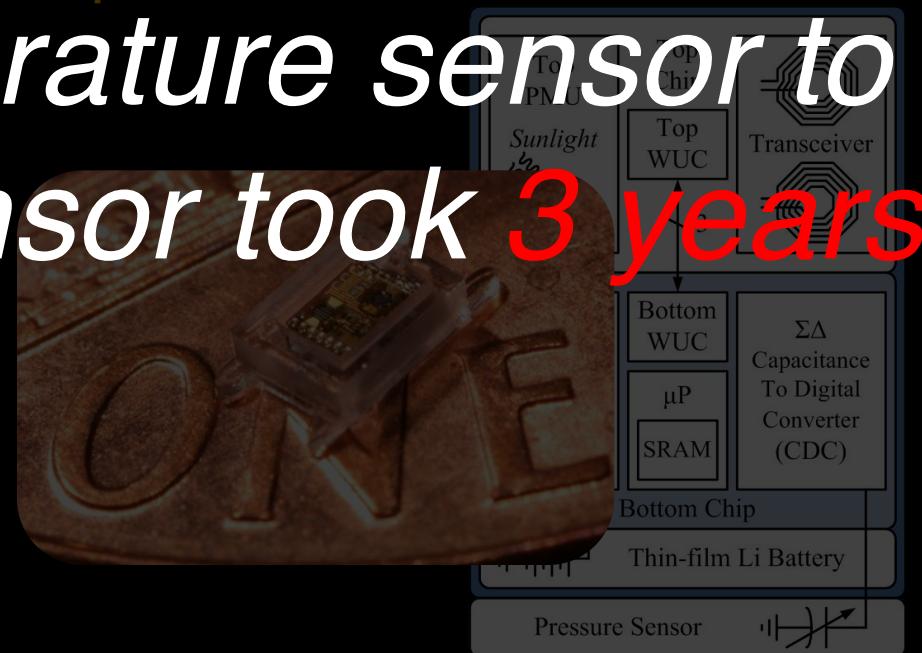
# Modularity is key for fast, iterative design, but was previously absent from mm-scale systems

- Phoenix 2008:
  - World's lowest power computer
  - Basically a **temperature sensor**
- Intraocular Pressure 2011:
  - Collaboration for glaucoma health
  - A **pressure sensor**

*From mm-scale temperature sensor to mm-scale pressure sensor took 3 years*

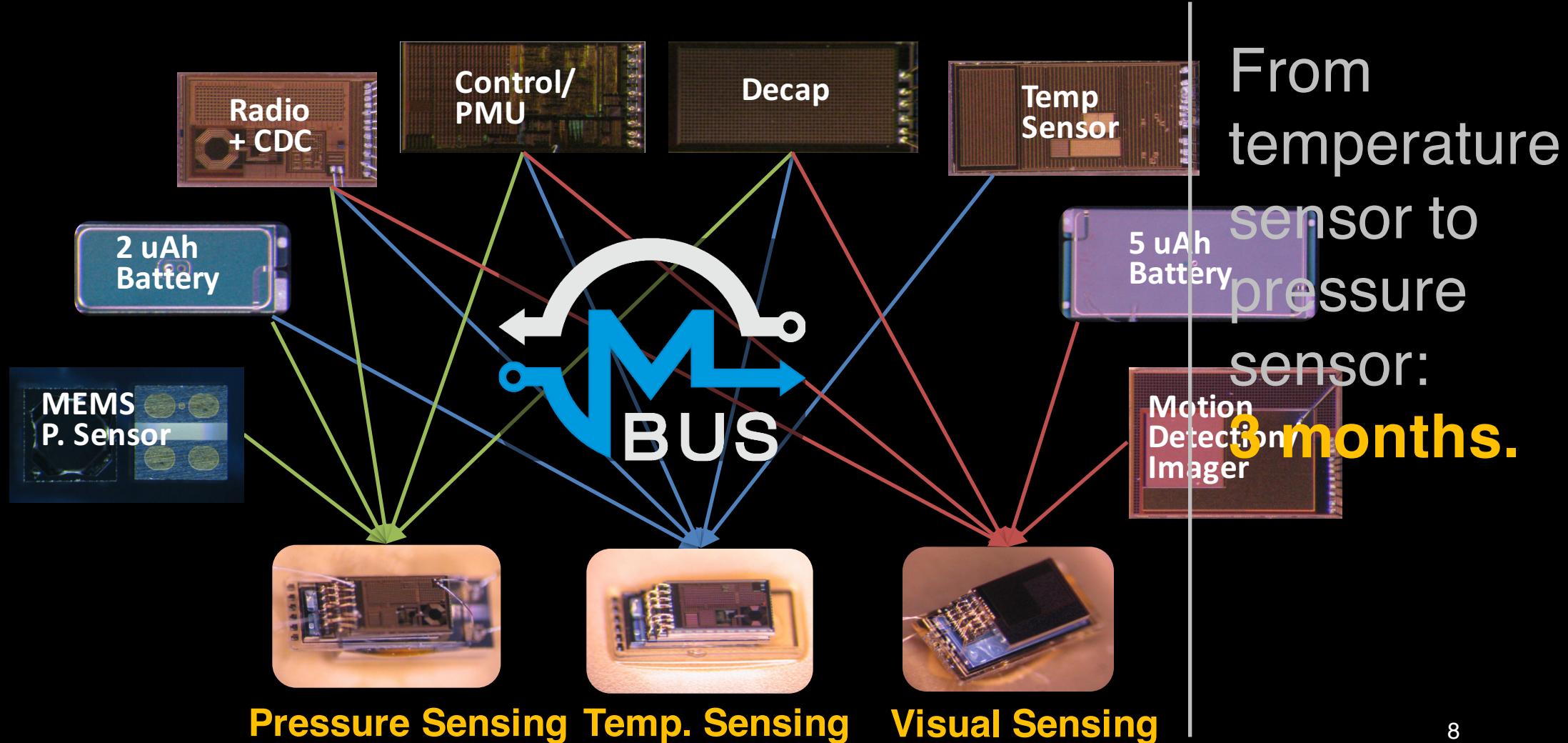


[The Phoenix Processor: A 30pW Platform for Sensor Applications](#), Mingoo Seok, Scott Hanson, Yu-Shiang Lin, Zhiyoong Foo, Daeyeon Kim, Yoonmyung Lee, Nurrachman Liu, Dennis Sylvester, David Blaauw , VLSI '08



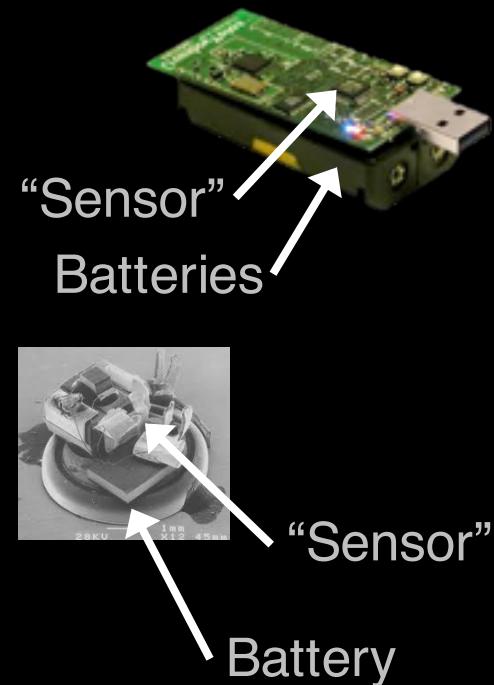
[A Cubic-Millimeter Energy-Autonomous Wireless Intraocular Pressure Monitor](#), Gregory Chen, Hassan Ghaed, Razi-ul Haque, Michael Wieckowski, Yejoong Kim, Gyouho Kim, David Fick, Daeyeon Kim, Mingoo Seok, Kensall Wise, David Blaauw, Dennis Sylvester, ISSCC '11

# MBus enables a modular, composable ecosystem of mm-scale components



# Existing interconnects have served us well for 30 years. What makes mm-scale systems unique?

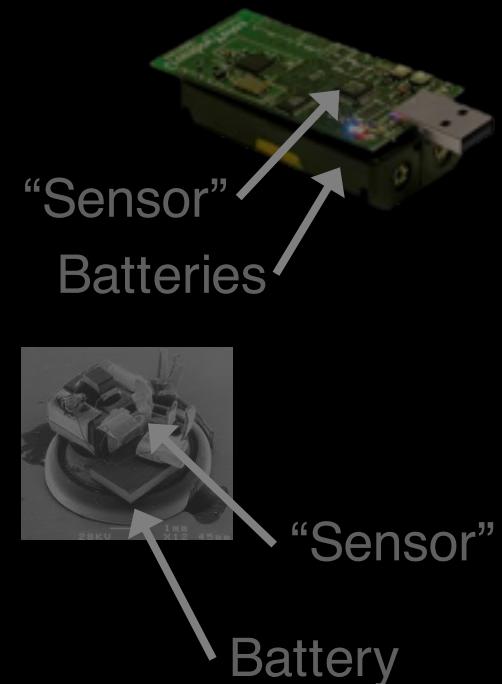
Node **volume** is dominated by **energy storage**



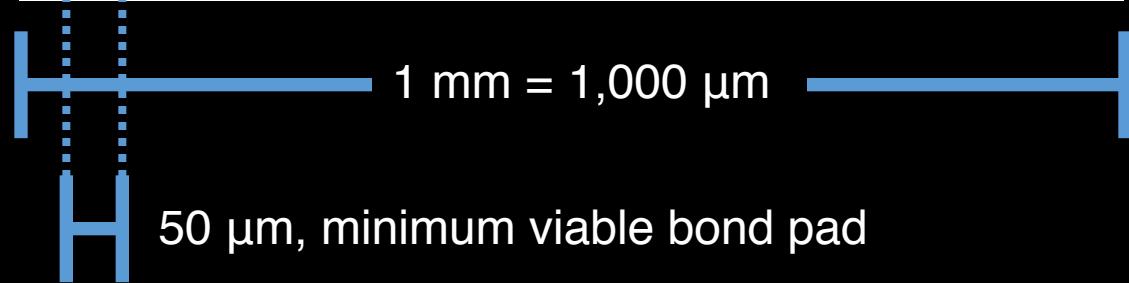
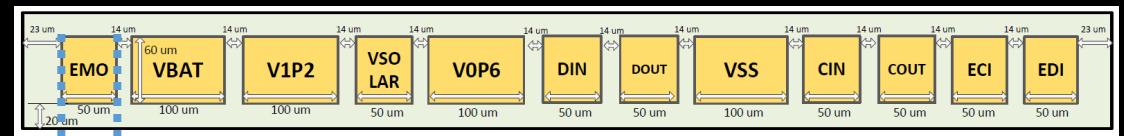
And volume is **shrinking cubically**  
10's  $\mu\text{W}$  active, 10's  $\text{nW}$  sleep, DC 0.1%

# Existing interconnects have served us well for 30 years. What makes mm-scale systems unique?

**Node volume** is dominated by energy storage



I/O pads begin to account for non-trivial percentage of node **surface area**



16-20 maximum I/O pins for 3D stacking

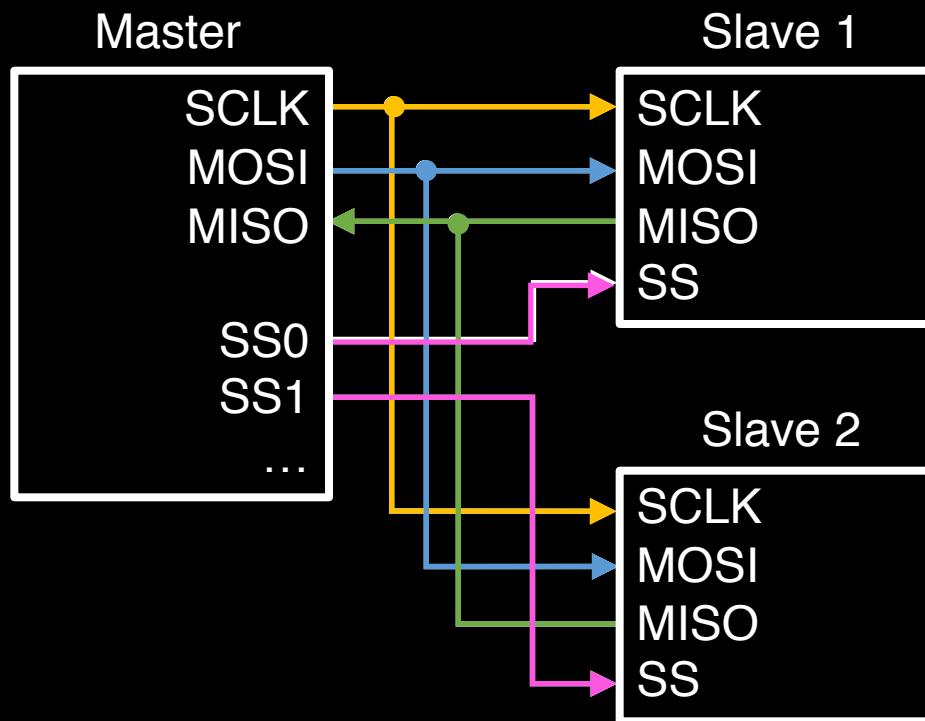
And volume is **shrinking cubically**  
10's  $\mu\text{W}$  active, 10's  $\text{nW}$  sleep, DC 0.1%

# SPI and I<sup>2</sup>C are like the USB and Firewire of embedded interconnects

- Nearly every microcontroller has both
- Nearly every peripheral has one or the other
- Very few use anything else
  - (except maybe UART)

# What is wrong with how are systems composed today?

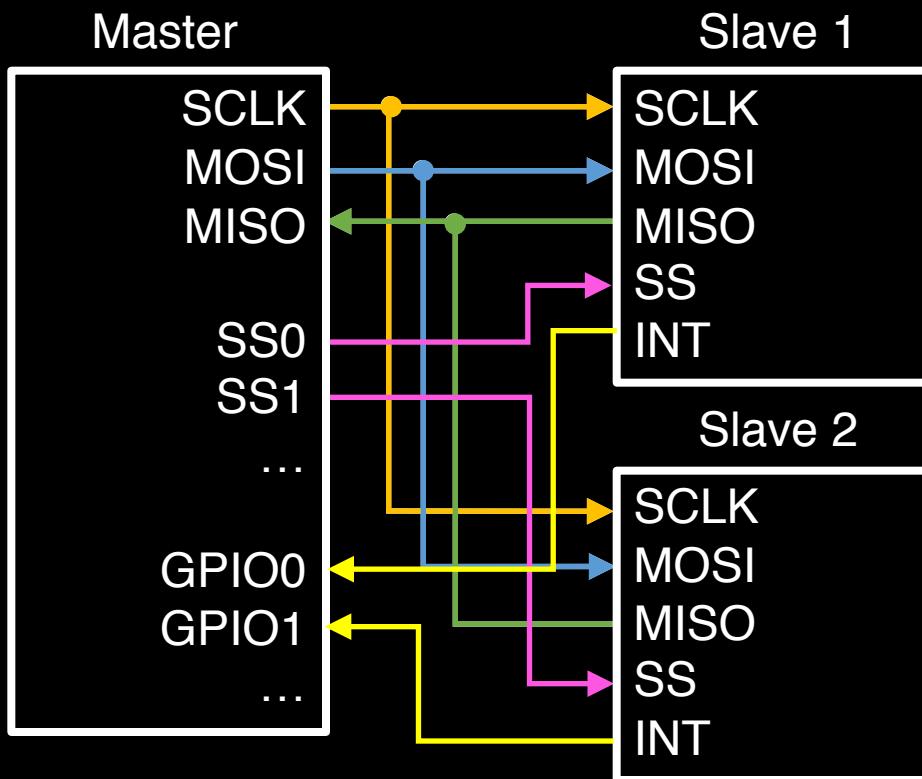
- SPI, invented by Motorola in ~1979



- One master, N slaves
- Shared clock: **SCLK**
- Shared data bus: **MOSI**
- Shared data bus: **MISO**
- One **Slave Select** line per slave
- **Key Properties**
  - One dedicated I/O line **per slave**
  - Master controls all communication

# What is wrong with how are systems composed today?

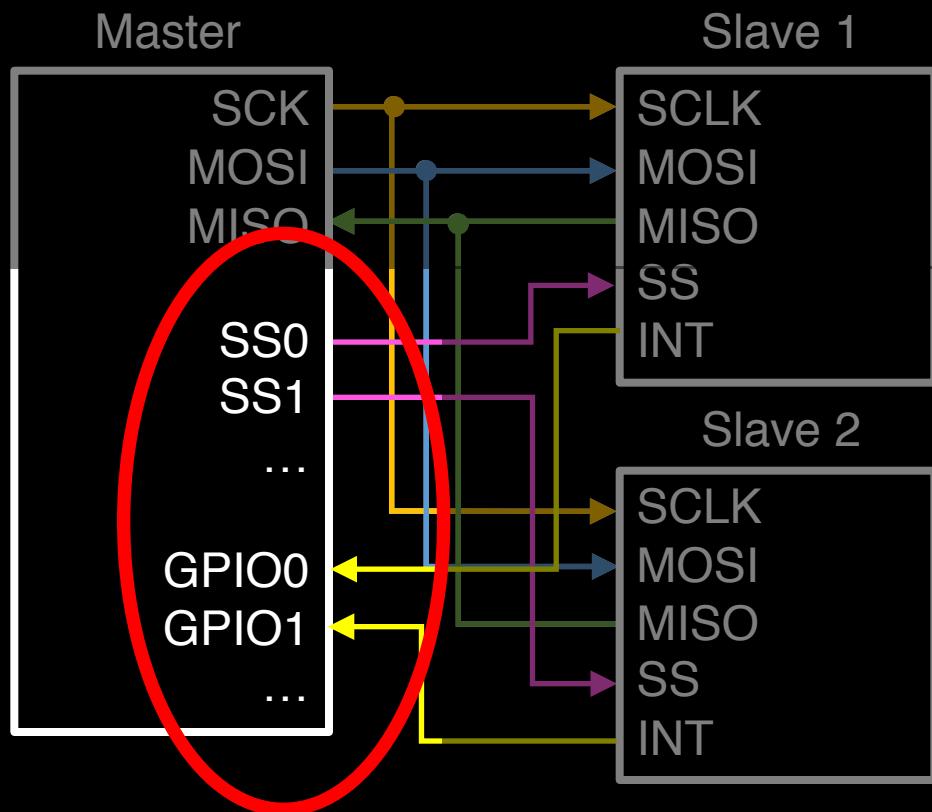
- SPI, invented by Motorola in ~1979



- One master, N slaves
- Shared clock: **SCLK**
- Shared data bus: **MOSI**
- Shared data bus: **MISO**
- One **Slave Select** line per slave
- One **interrupt line** per slave
- **Key Properties**
  - ~~One~~ Two dedicated I/O lines **per slave**
  - Master controls all communication
    - Interrupts must be out-of-band

# SPI's I/O overhead and centralized architecture **do not scale** to mm-scale systems

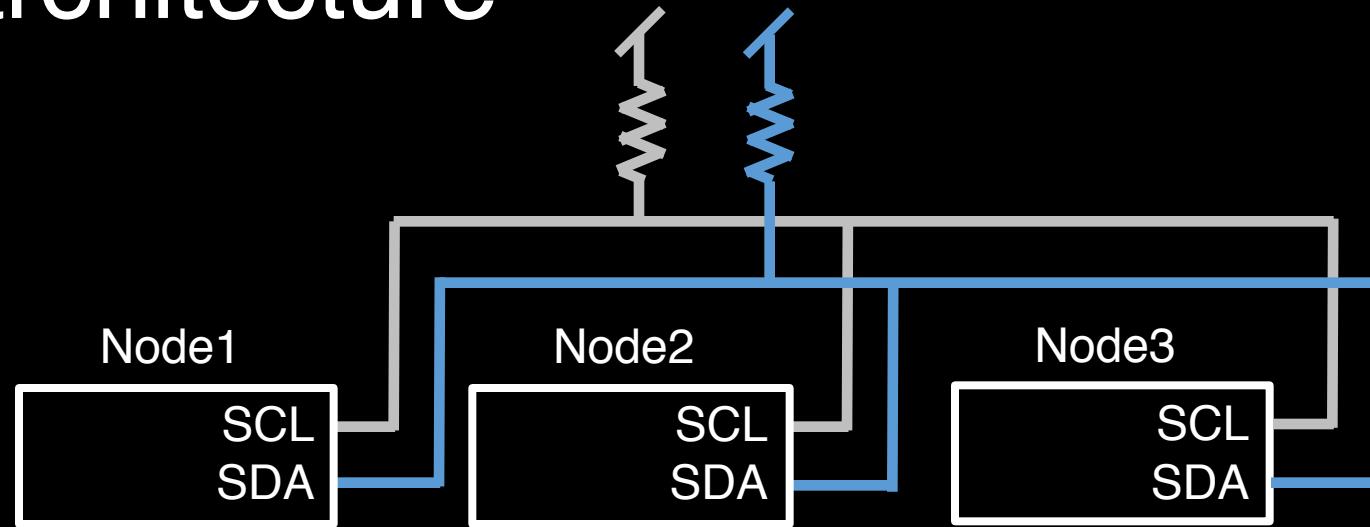
- SPI, invented by Motorola in ~1979



- One master, N slaves
- Shared clock: **SCLK**
- Shared data bus: **MOSI**
- Shared data bus: **MISO**
- One **Slave Select** line per slave
- One **interrupt line** per slave
- **Key Properties**
  - ~~One~~ Two dedicated I/O lines **per slave**
  - Master controls all communication
    - Interrupts must be out-of-band

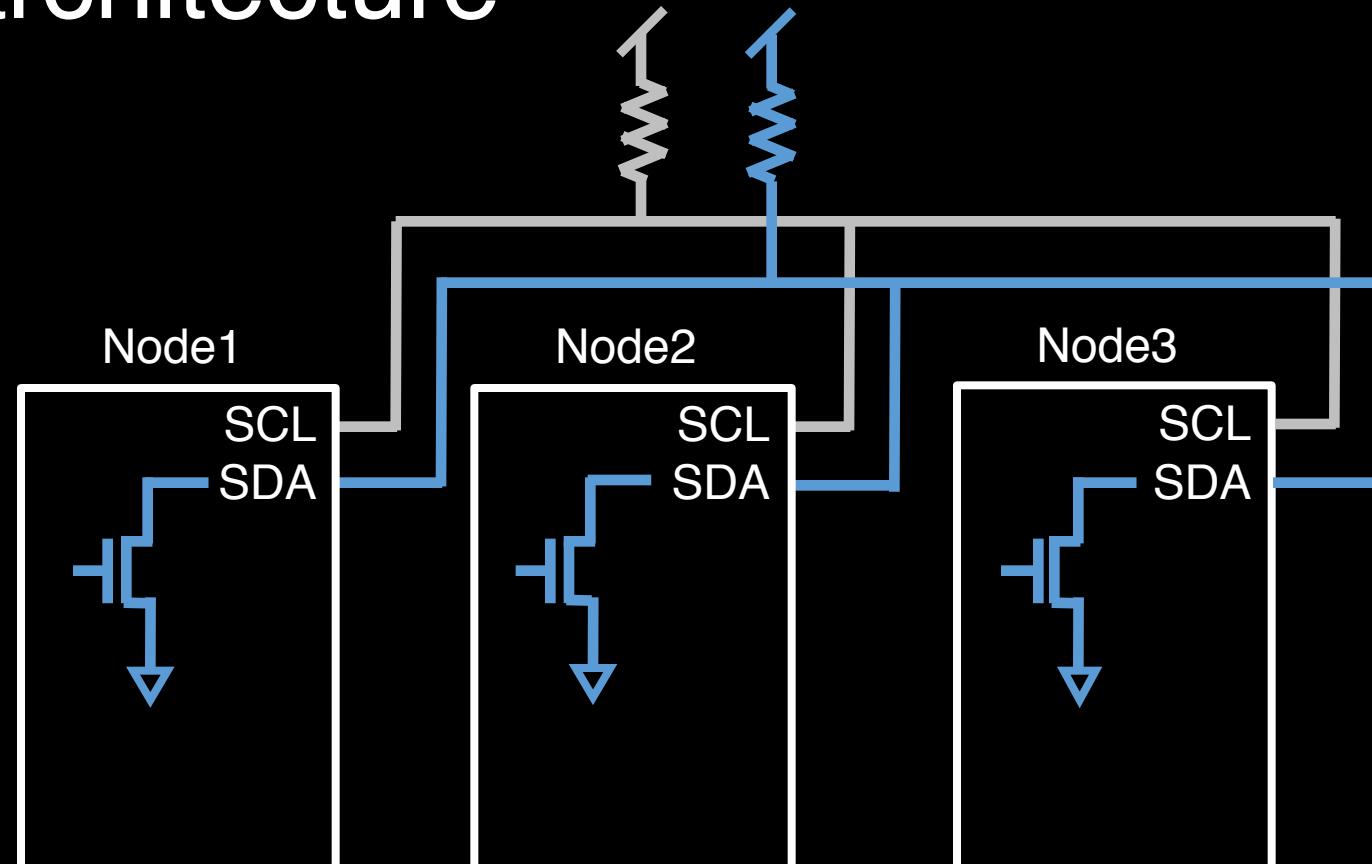
# I<sup>2</sup>C has fixed I/O requirements and a decentralized architecture

- I<sup>2</sup>C, invented by Phillips in 1982
  - Any-to-(m)any on one shared bus
- **Key Properties**
  - Fixed wire count (2)



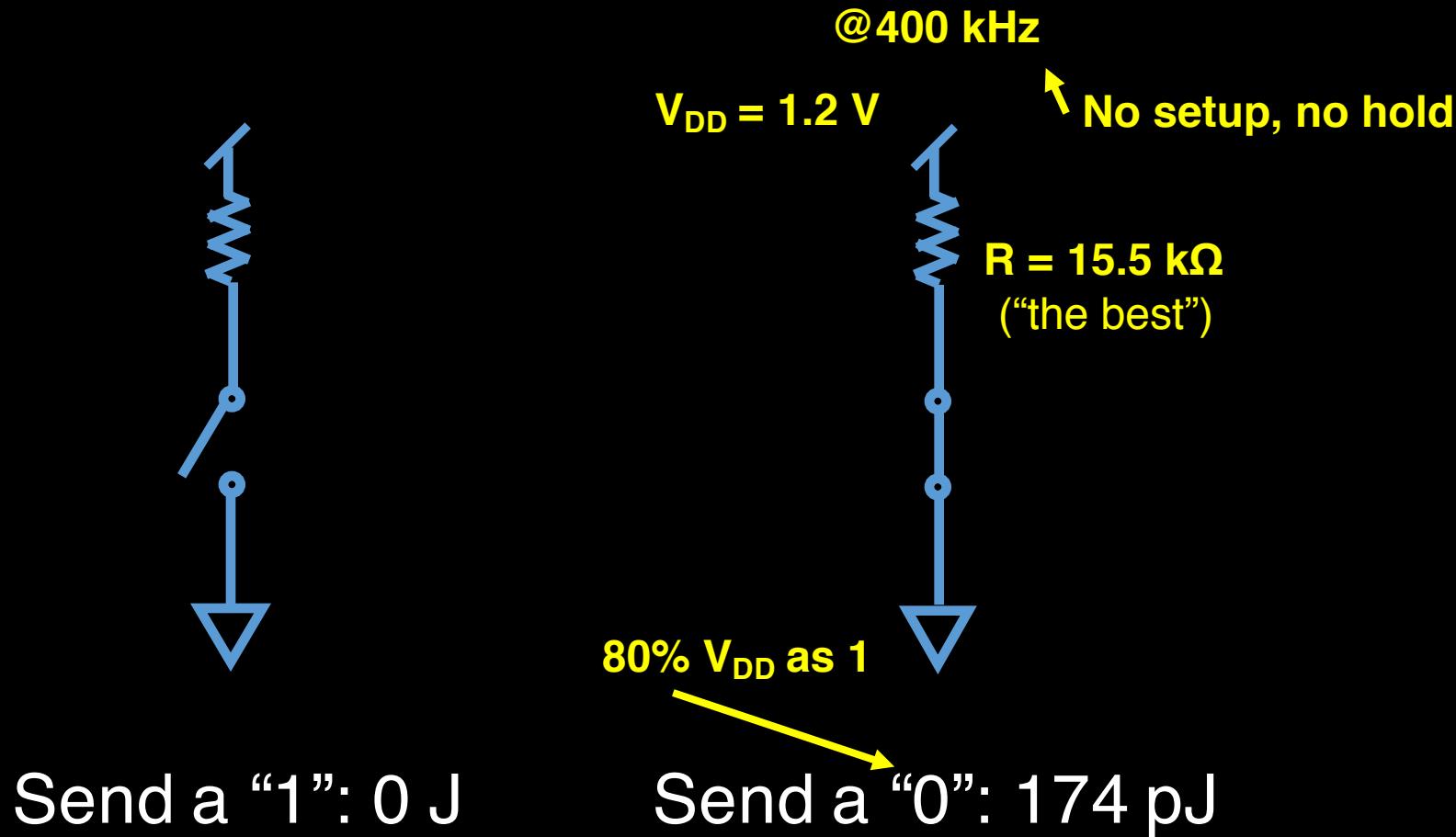
# I<sup>2</sup>C has fixed I/O requirements and a decentralized architecture

- I<sup>2</sup>C, invented by Phillips in 1982
  - Any-to-(m)any on one shared bus
- **Key Properties**
  - Fixed wire count (2)
  - Open-collector
    - Multi-master
    - Flow Control

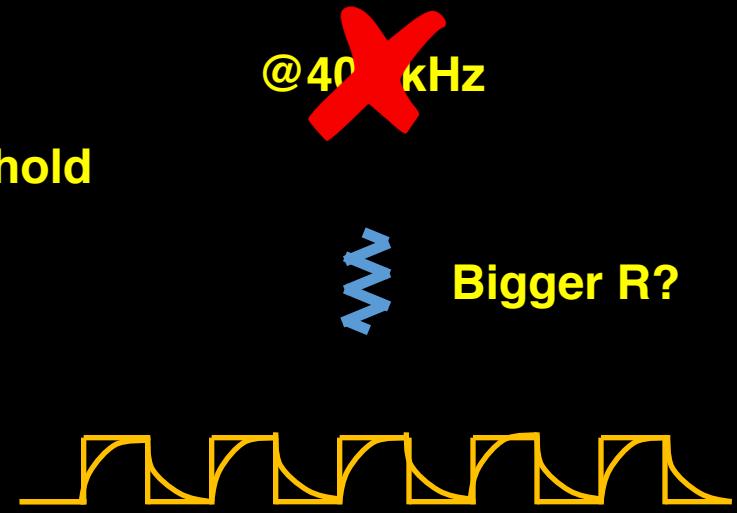
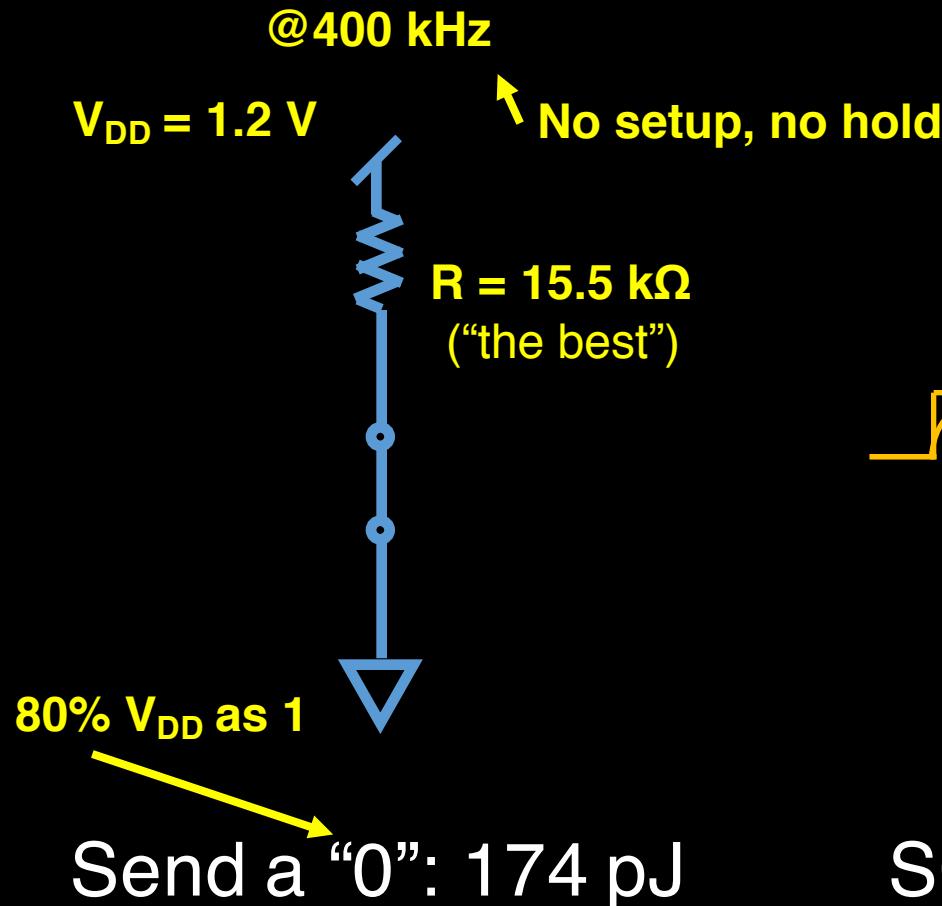
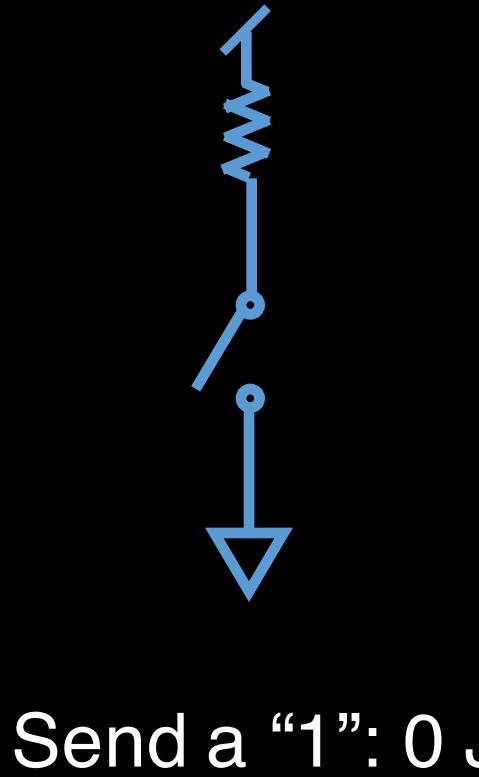


*Open-collector (aka wired-AND)*

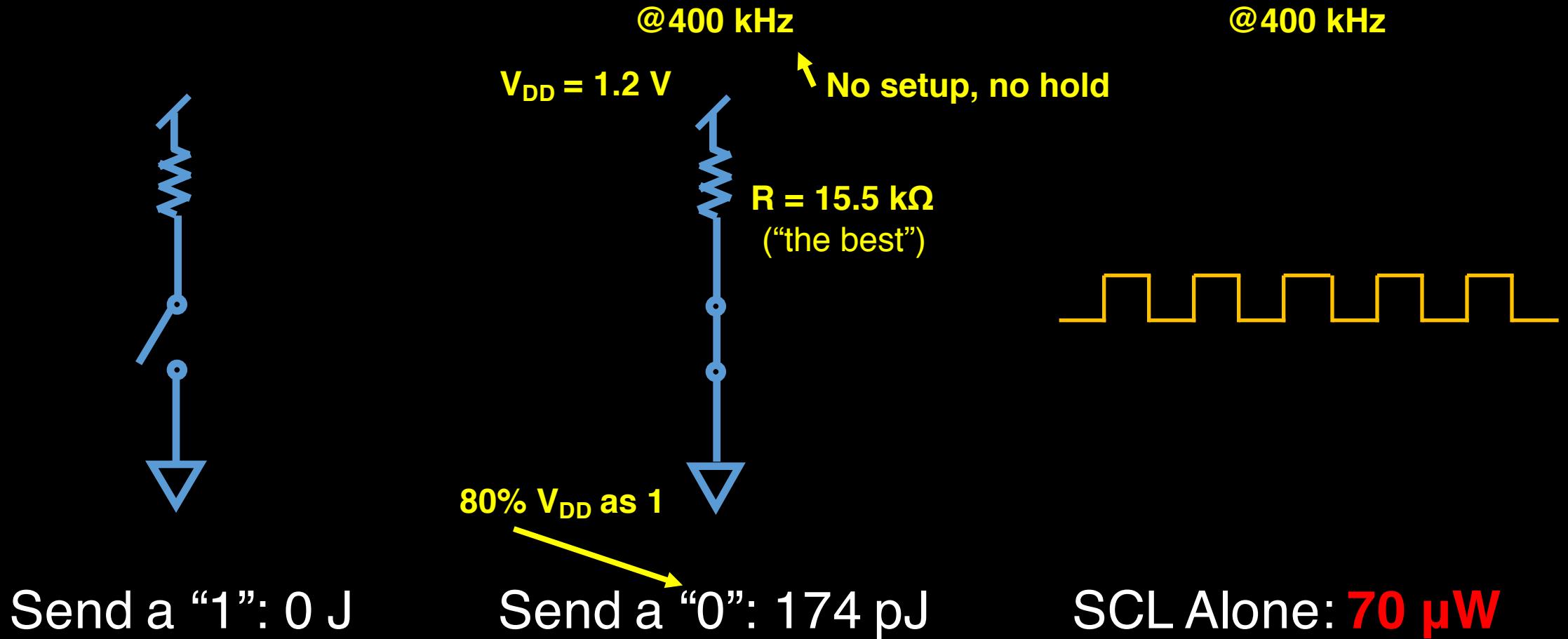
The problem is the energy costs of running an open-collector



The problem is the energy costs of running an open-collector



The problem is the energy costs of running an open-collector

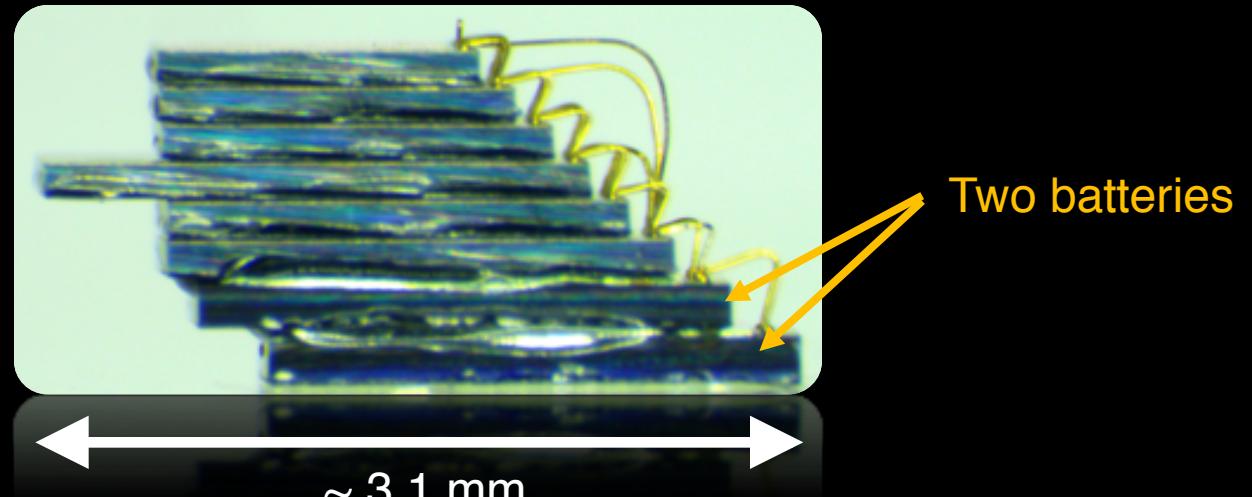


# The energy demands of open-collectors make them unsuitable for mm-scale systems

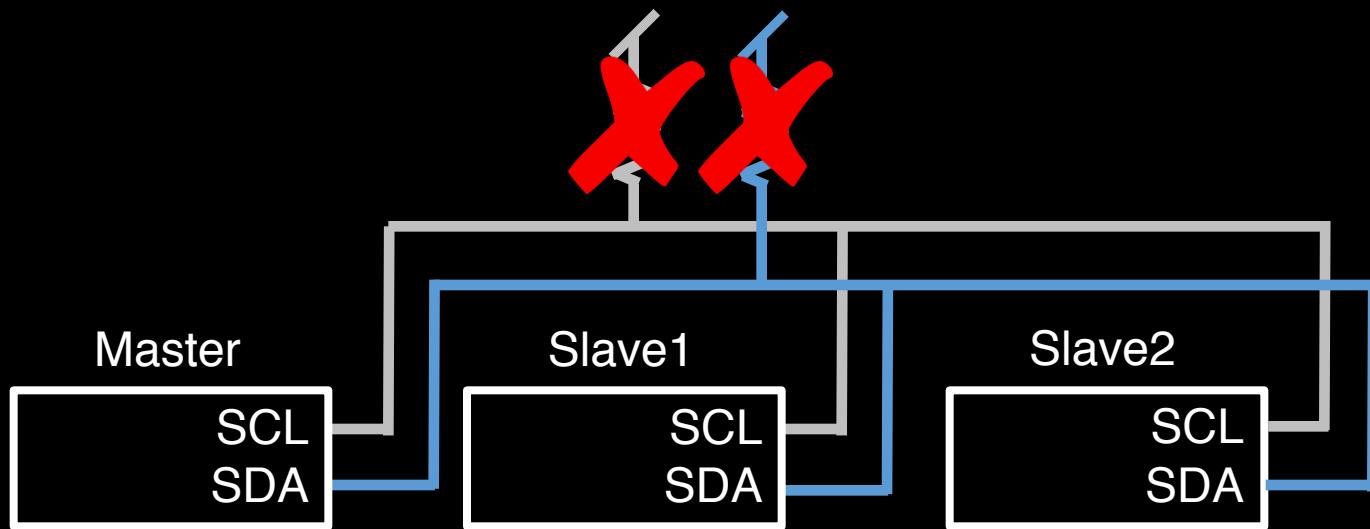
- Active Energy Budget: **20  $\mu\text{W}$**
- Not an arbitrary number:
  - Volume Target
  - Lifetime Target



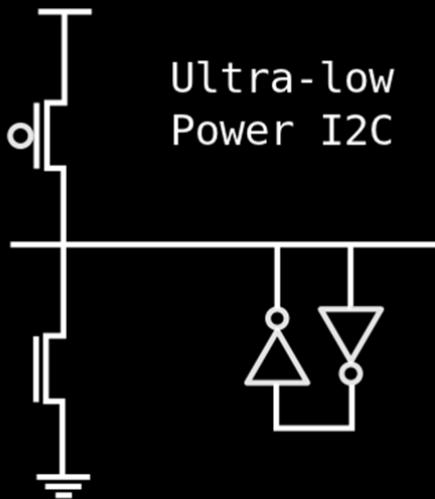
SCL Alone: **70  $\mu\text{W}$**



Can we modify I<sup>2</sup>C to bring energy costs in line with mm-scale?



# Replace the passive pull-up resistor with active circuitry



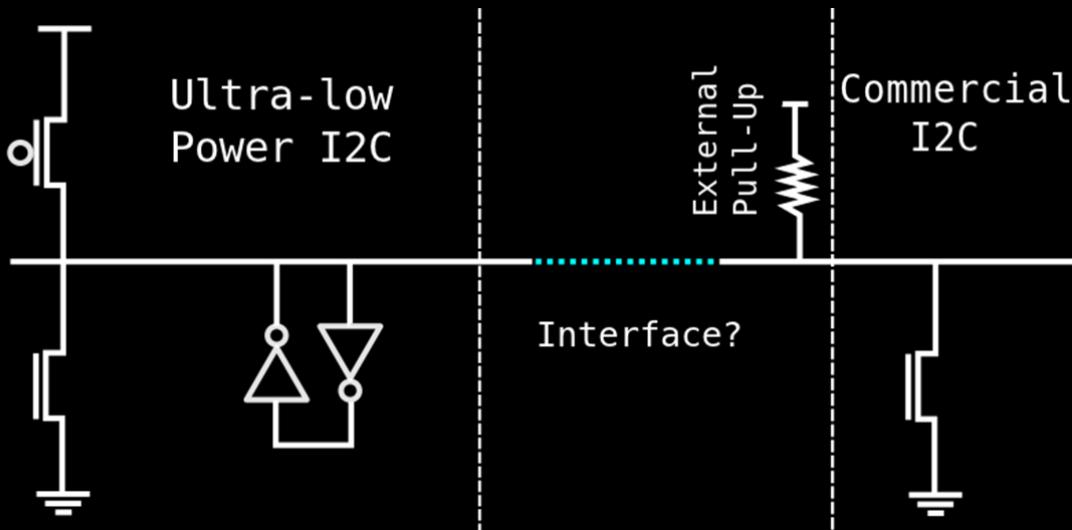
[A Modular 1 mm<sup>3</sup> Die-Stacked Sensing Platform with Low Power I<sub>2</sub>C Inter-die Communication and Multi-Modal Energy Harvesting](#)

Yoonmyung Lee, Suyoung Bang, Inhee Lee, Yejoong Kim, Gyouho Kim, Mohammed Hassan Ghaed, Pat Pannuto, Prabal Dutta, Dennis Sylvester, David Blaauw  
IEEE Journal of Solid-State Circuits

The Good : Able to achieve **88 pJ / bit (measured)**

The Bad : Required clocks running at **5x bus clock** on every chip

# Replace the passive pull-up resistor with active circuitry



## Ultra-Constrained Sensor Platform Interfacing

Pat Pannuto, Yoonmyung Lee, Benjamin Kempke,  
Dennis Sylvester, David Blaauw, and Prabal Dutta  
IPSN 2012 (Demo)

The Good : Able to achieve **88 pJ / bit (measured)**

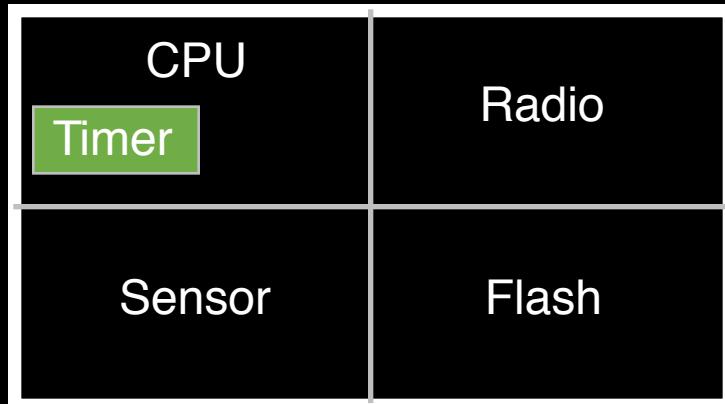
The Bad : Required clocks running at **5x bus clock** on every chip

The Bad : “I<sup>2</sup>C-like” is not I<sup>2</sup>C – required FPGA to integrate with COTS chips

The Ugly : Hand-tuned, ratioed logic on every chip – **Not synthesizable**

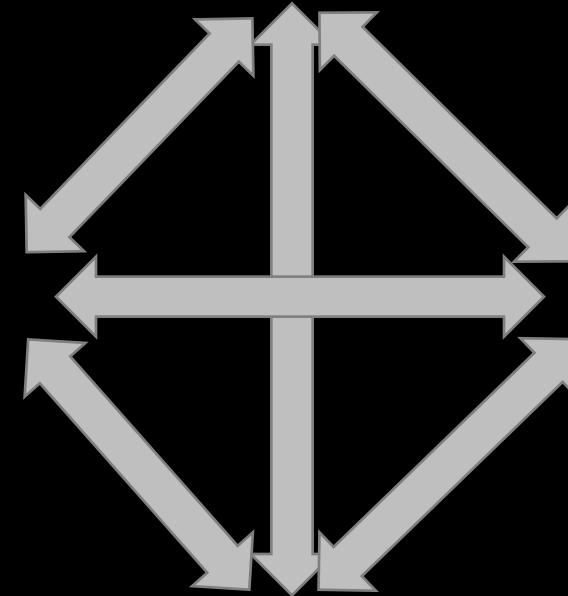
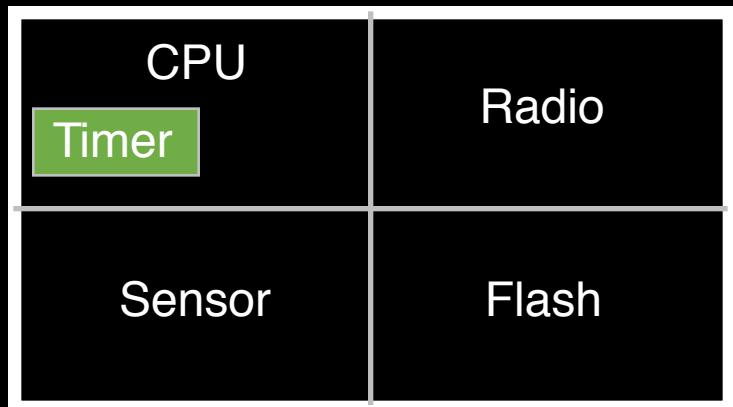
# “Dark silicon” is more like “dimly lit silicon”

- Clock-gated modules still exhibit static leakage
  - Blows mm-scale power budget
- mm-scale systems perform **power-gating**
  - This means modules are cold-booting all the time
- Manageable for monolithic designs because *something* always powered on

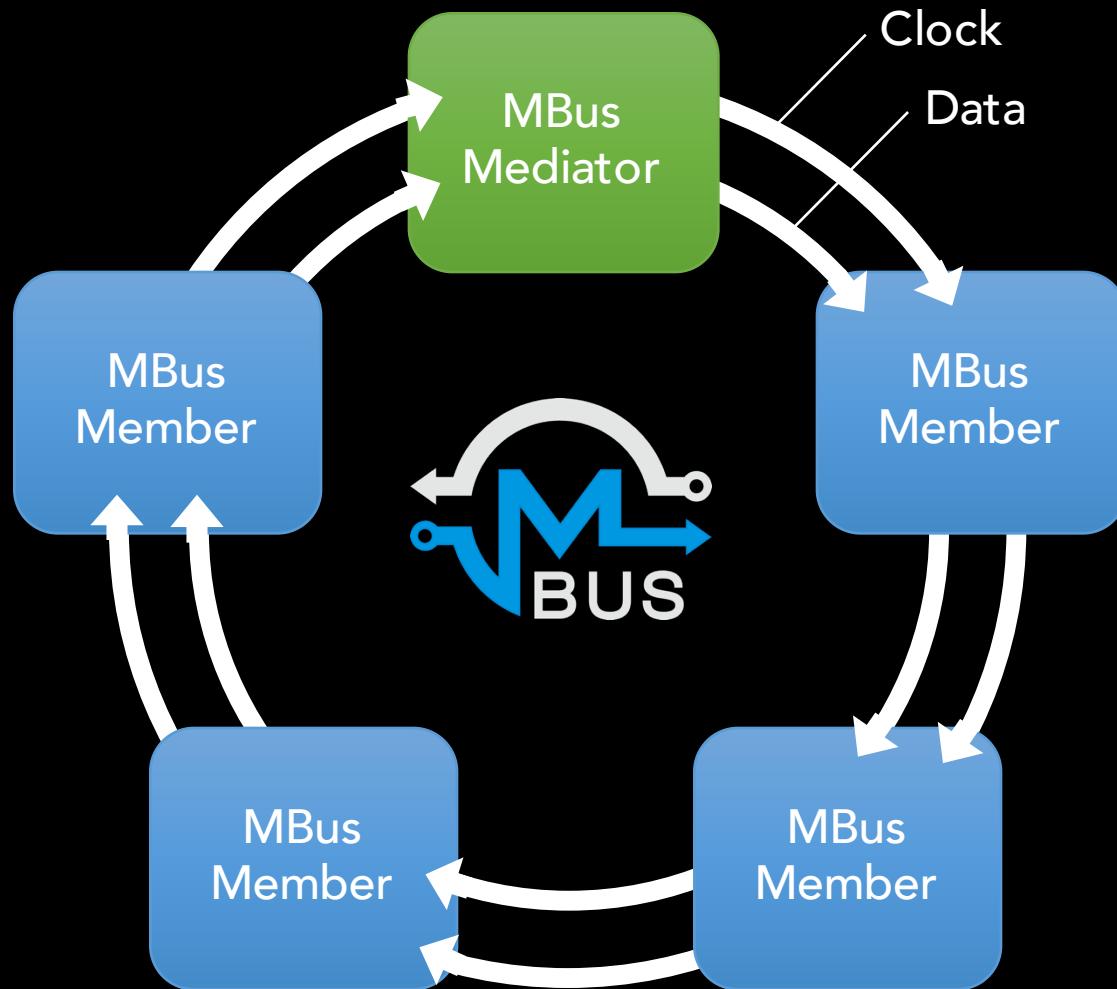


# Modular mm-scale components introduce novel circuits problems and novel systems problems

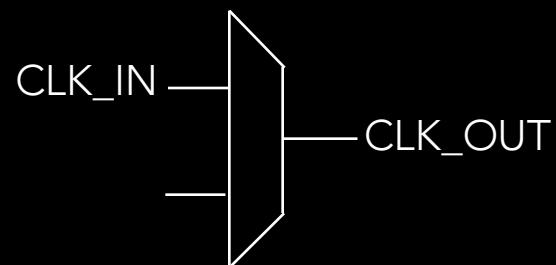
- Clockless cold boot circuits are tricky
- How do you know what's awake?
- How do you communicate with “pitch black” silicon to wake it up?
  - I<sup>2</sup>C-variant: custom “wakeup” signal



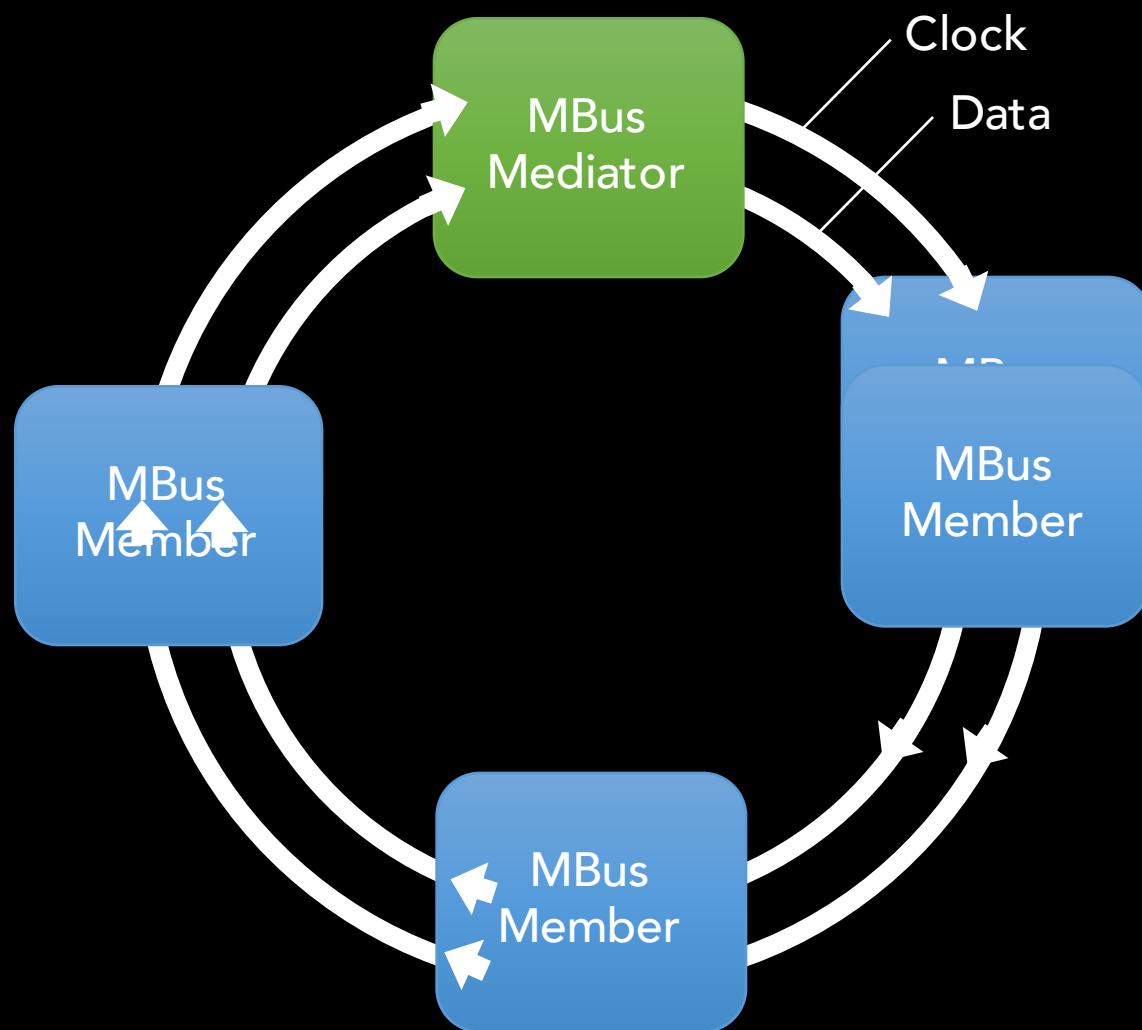
The MBus design follows from a careful consideration of all the requirements for modular, mm-scale systems



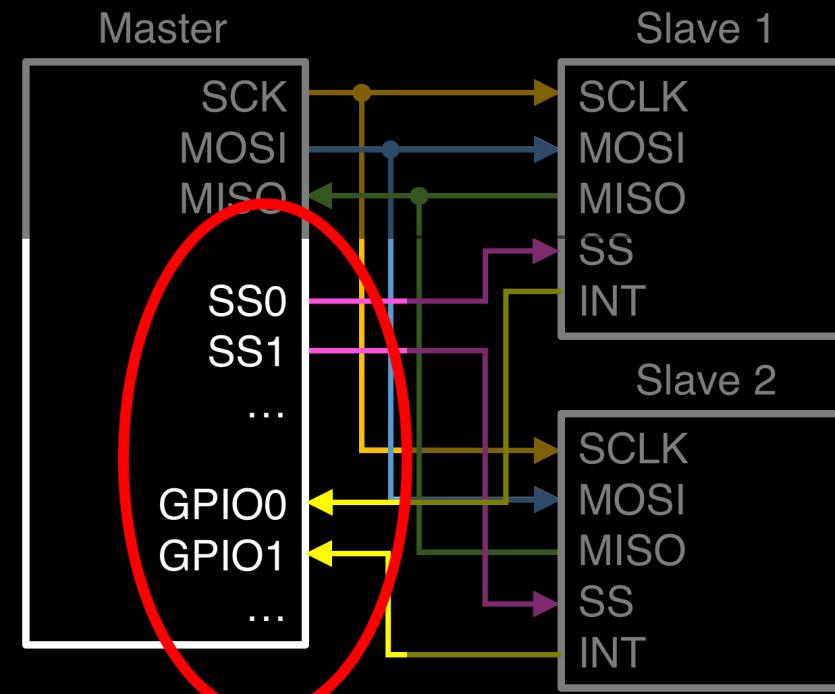
- Ring Topology
- 2 lines – 4 I/O per node
  - Clock
  - Data
- “Shoot-Through”



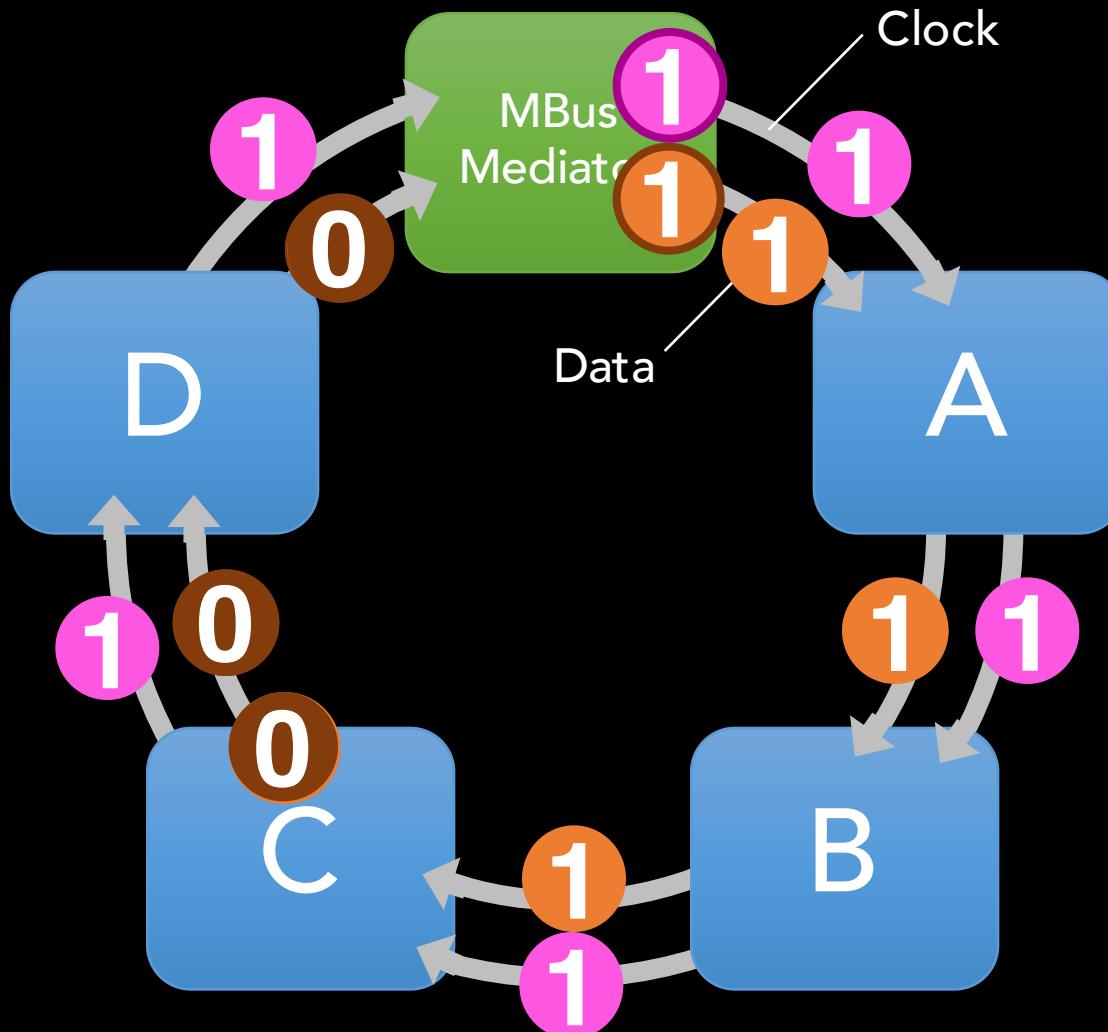
To be extensible and respect I/O constraints, wire count must be independent of node count



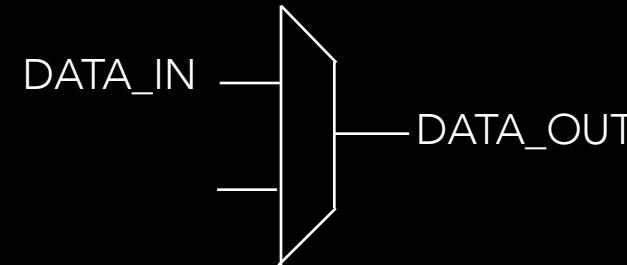
*Recall SPI:*



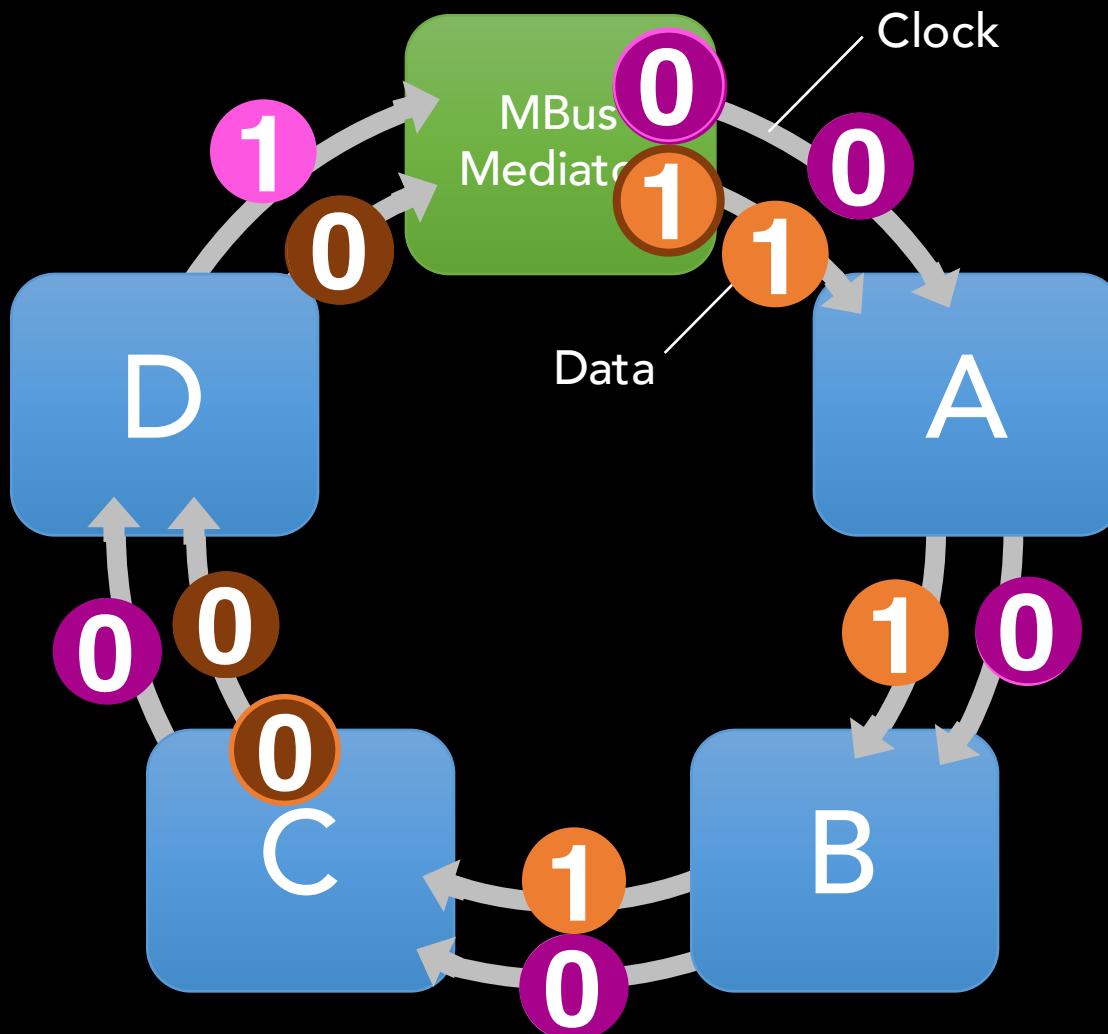
# Supporting interrupts with a fixed number of single-ended connections requires an arbitration protocol



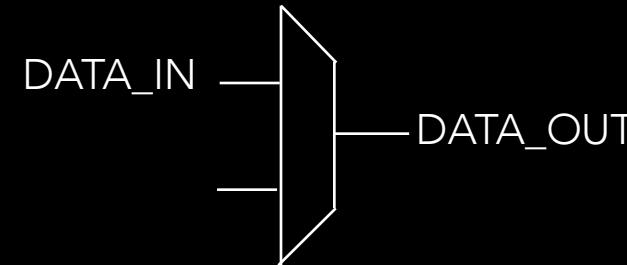
- Recall: “shoot through”
- C wants to send a message
  - Stop forwarding, drive 0
- The **mediator** does not forward during arbitration
  - Also generates the bus clock



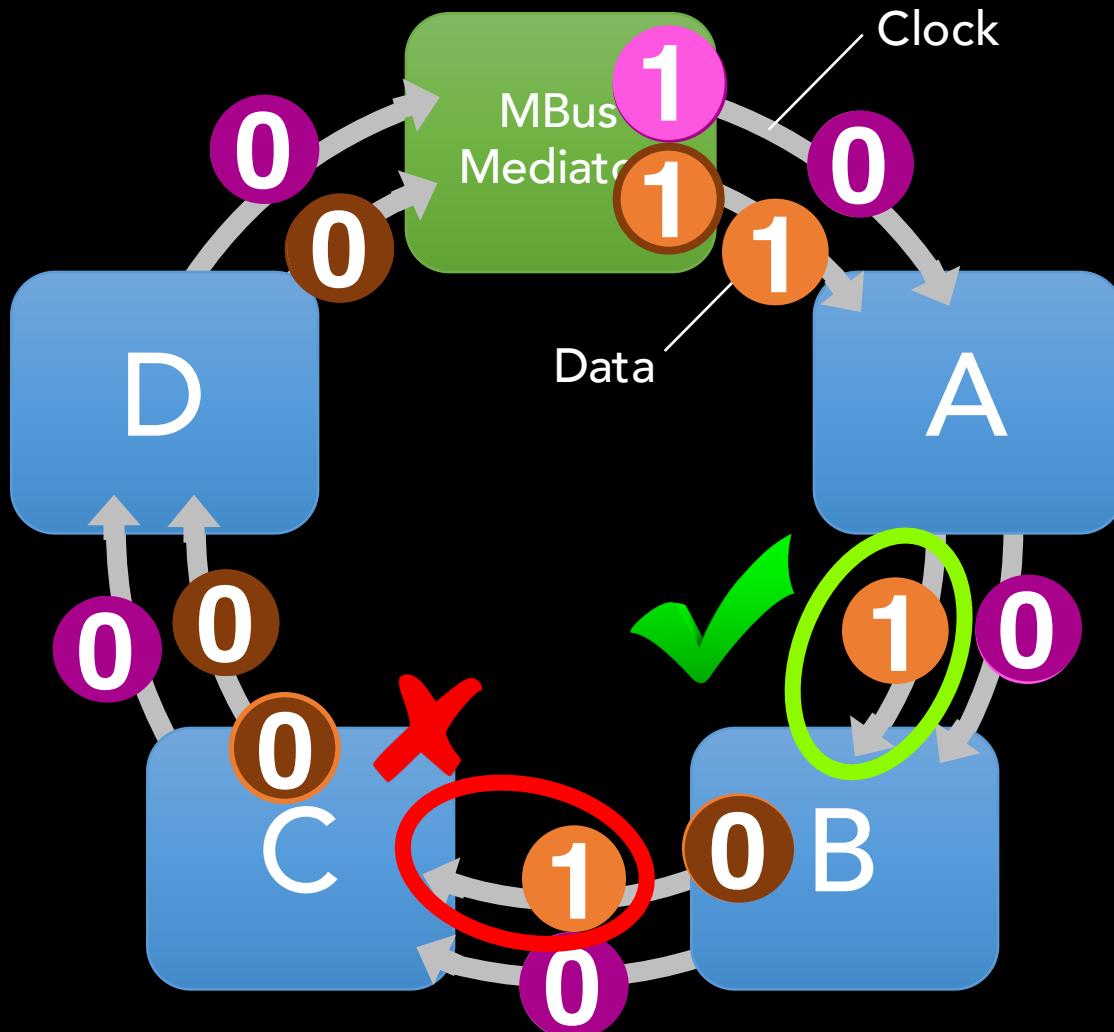
# Supporting interrupts with a fixed number of single-ended connections requires an arbitration protocol



- Recall: “shoot through”
- C wants to send a message
  - Stop forwarding, drive 0
- The **mediator** does not forward during arbitration
  - Also generates the bus clock



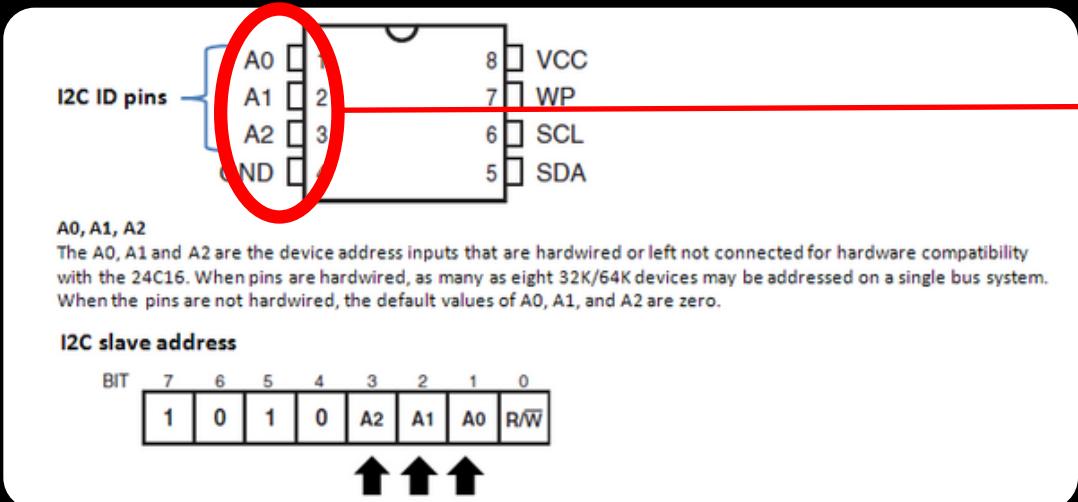
# Supporting interrupts with a fixed number of single-ended connections requires an arbitration protocol



- What changes if B tries to send as well as C?
  - B and C drive DATA\_OUT to 0
- B's DATA\_IN high, wins
- C's DATA\_IN low, loses
- MBus has topological priority

# Tradeoff between globally unique addresses, address length, and overhead

- Embed addresses in message frames
  - Overhead proportional to number of uniquely addressable device
- I<sup>2</sup>C uses 7-bit device addresses with design-time LSBs



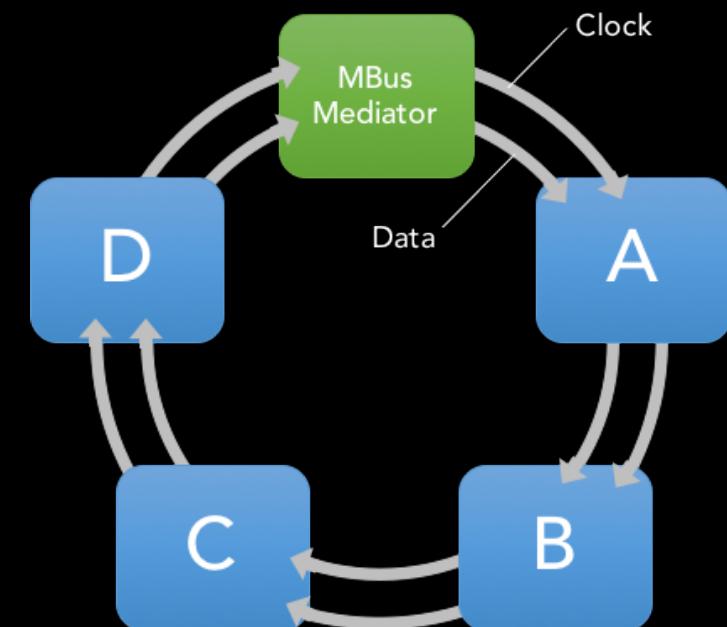
- Requires I/O not available
- Makes packaging assumptions
  - mm-scale systems not always PCB
  - Routing may not be easy
    - 3D stack
    - Flip-chip + TSVs

# Tradeoff between globally unique addresses, address length, and overhead

- 3 Options
  - Short static addresses and allow device conflicts
  - Long static addresses to avoid device conflicts
  - Non-static addresses
- MBus does all 3
  - 4-bit: Static short prefixes (device class)
  - 24-bit: Static long prefixes (unique device ID)
  - 4-bit: Runtime enumeration protocol (replaces short prefix)

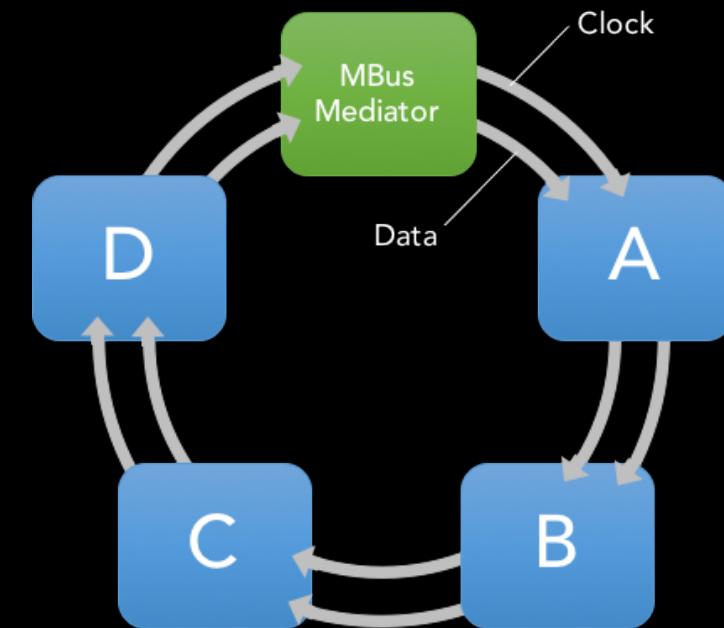
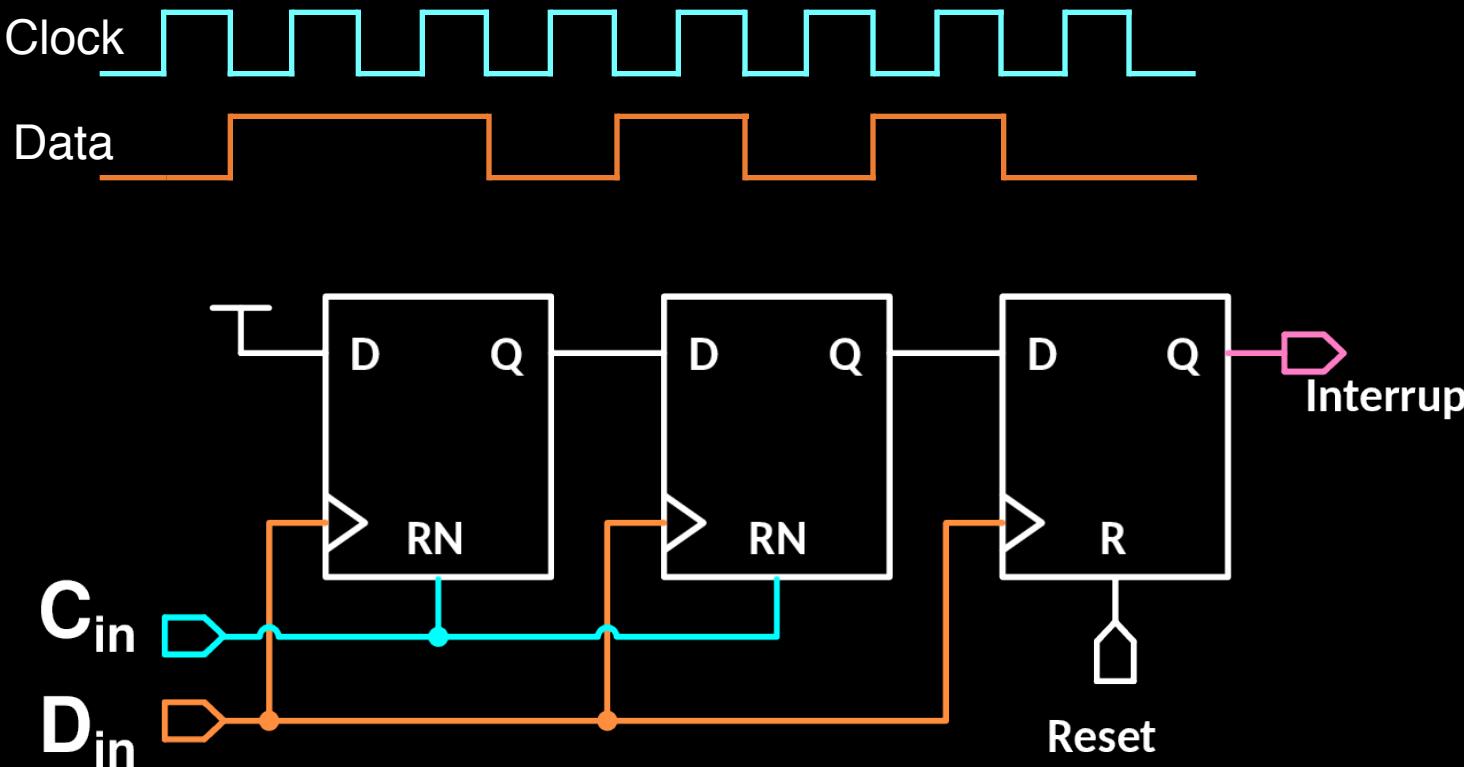
# Unbounded messages maximize flexibility and minimize overhead

- An MBus message is 0...N bytes of data
- Embed length in message
  - Imposes large overhead for short messages
  - Forces fragmentation of long messages
- “End-of-message” sentinel byte(s)
  - Imposes large overhead for short messages
  - Requires escaping if sentinel is in transmitted
  - Data-dependent behavior, hard to reason about
    - Worst case 2x overhead!



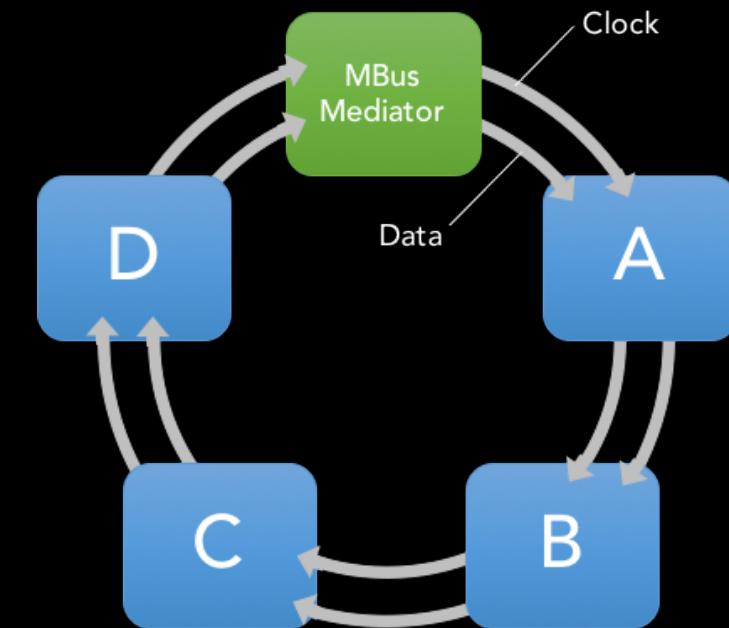
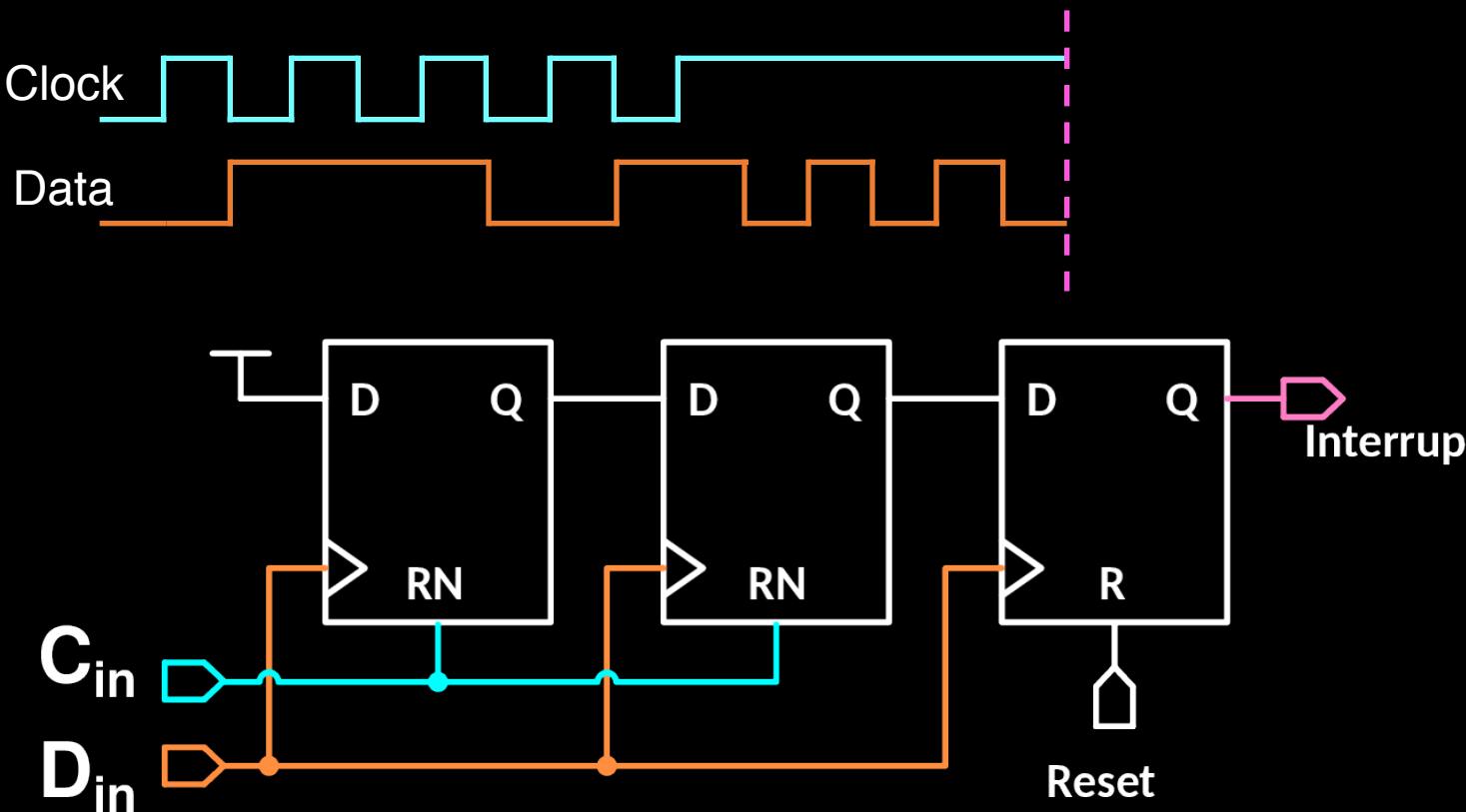
# MBus “interjections” provide an in-band end-of-message with minimal overhead

- During normal operation, Data toggles slower than Clock



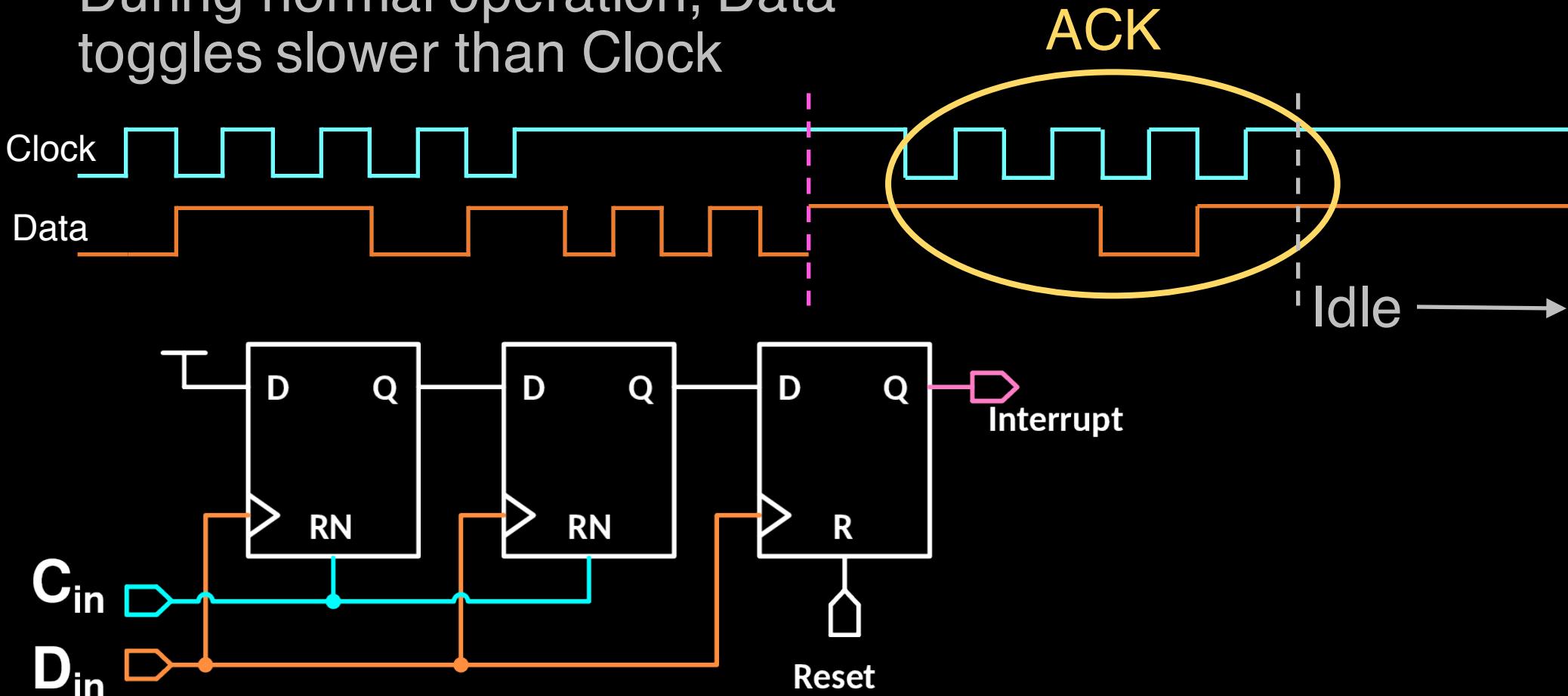
# MBus “interjections” provide an in-band end-of-message with minimal overhead

- During normal operation, Data toggles slower than Clock

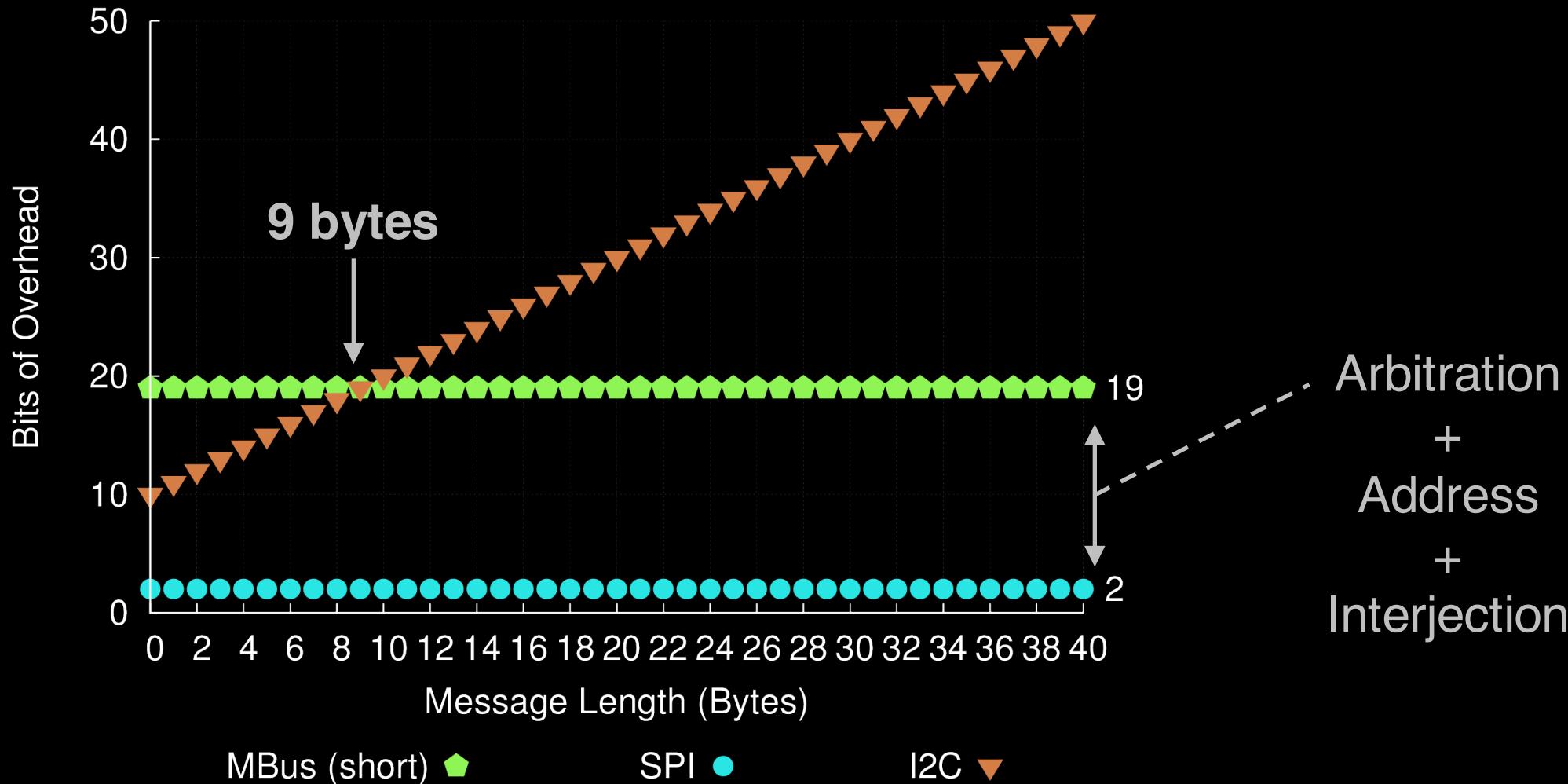


# MBus “interjections” provide an in-band end-of-message with minimal overhead

- During normal operation, Data toggles slower than Clock

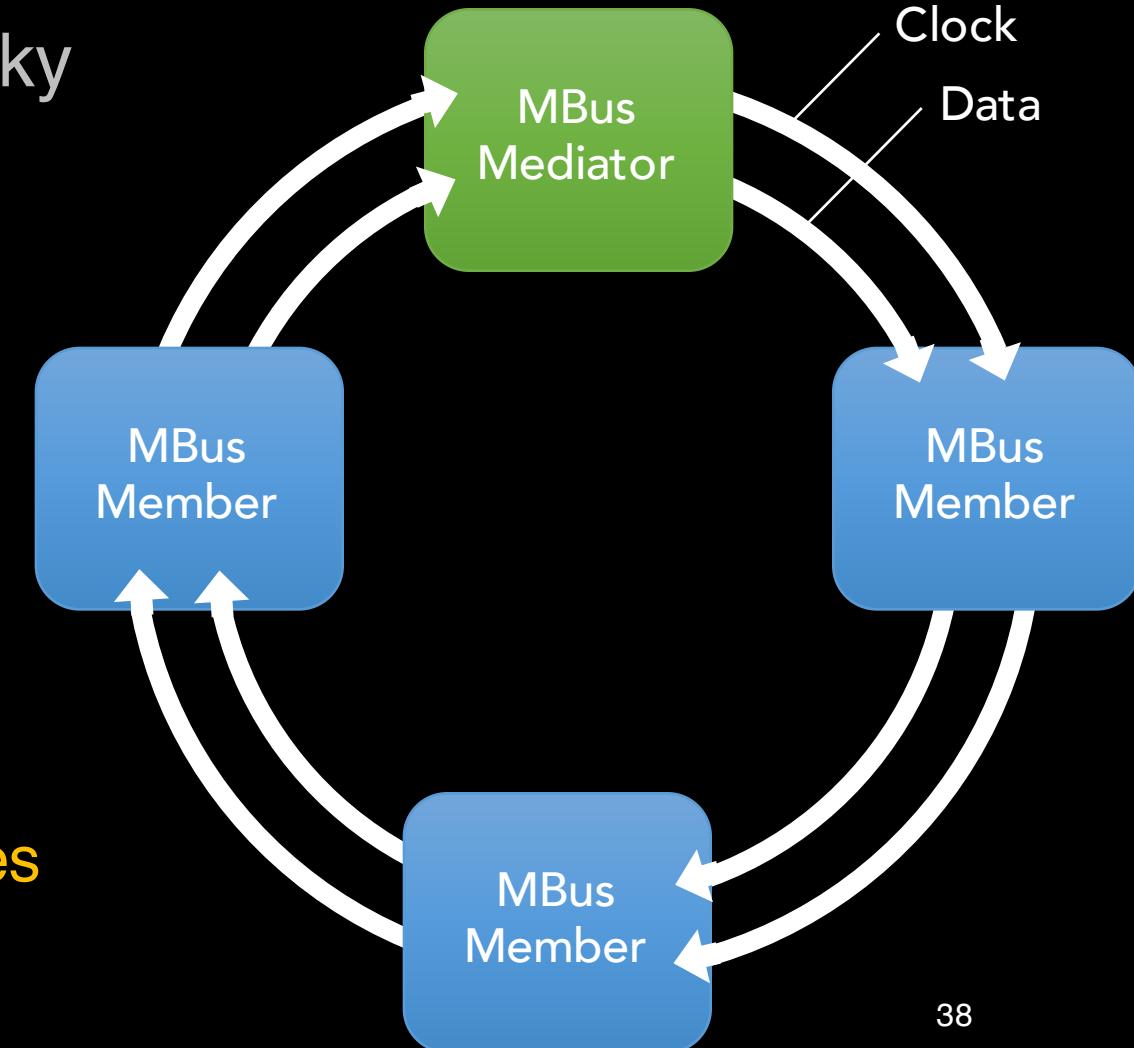
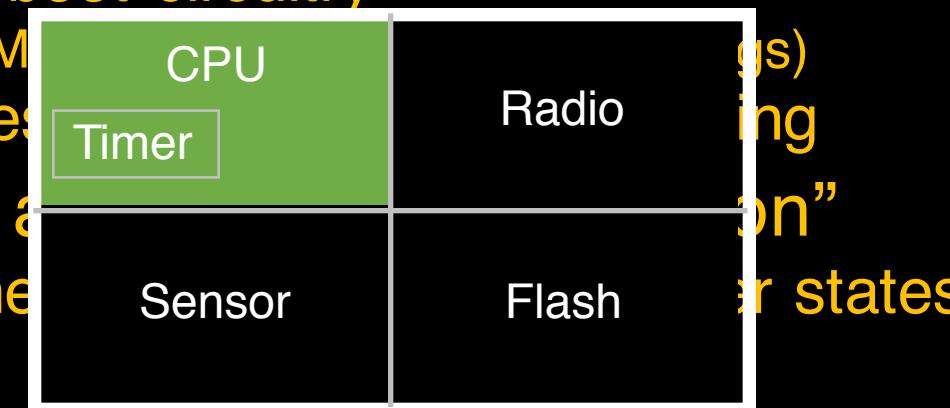


# Transaction-level ACKs minimize common-case overhead while interjections preserve flow control



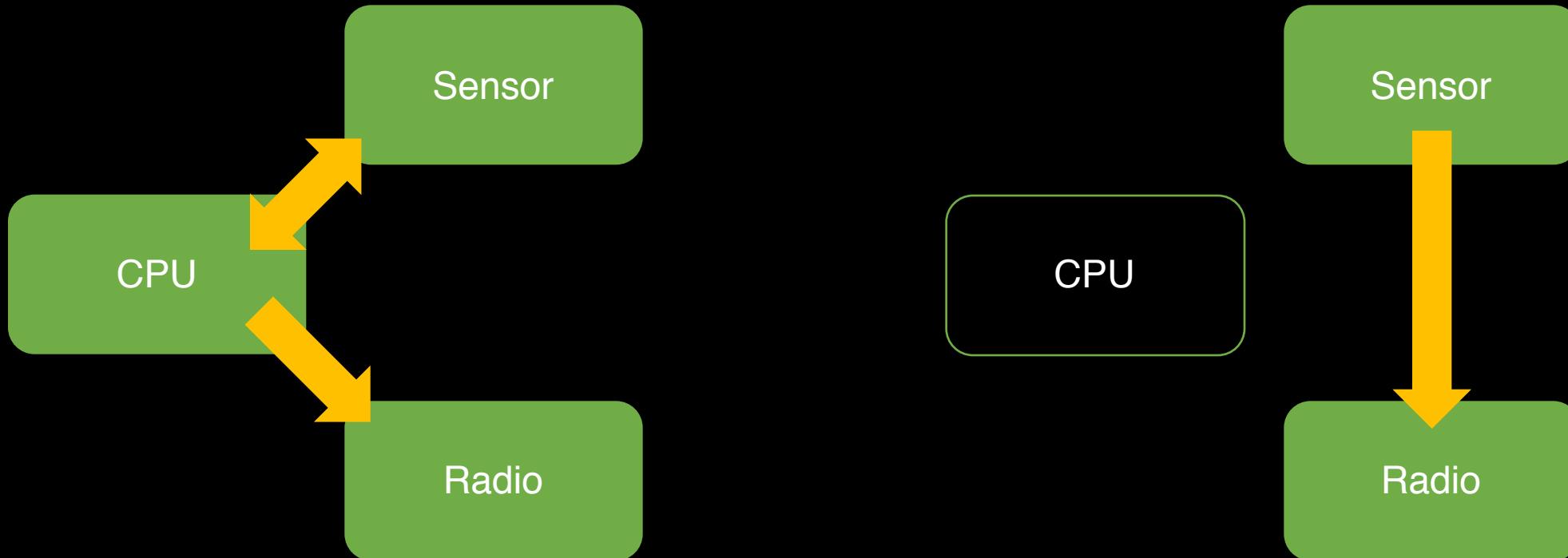
# The modularity enabled by MBus created a circuits problem and a systems problem

- Clockless cold boot circuits are tricky
- How do you communicate with “pitch black” silicon to wake it up?
  - How do you know what’s awake?
- A power-gated node cannot send
  - Use arbitration edges to drive the cold-boot circuitry
    - (M)Nodes
- Nodes are “on” or “off”
  - No need for states



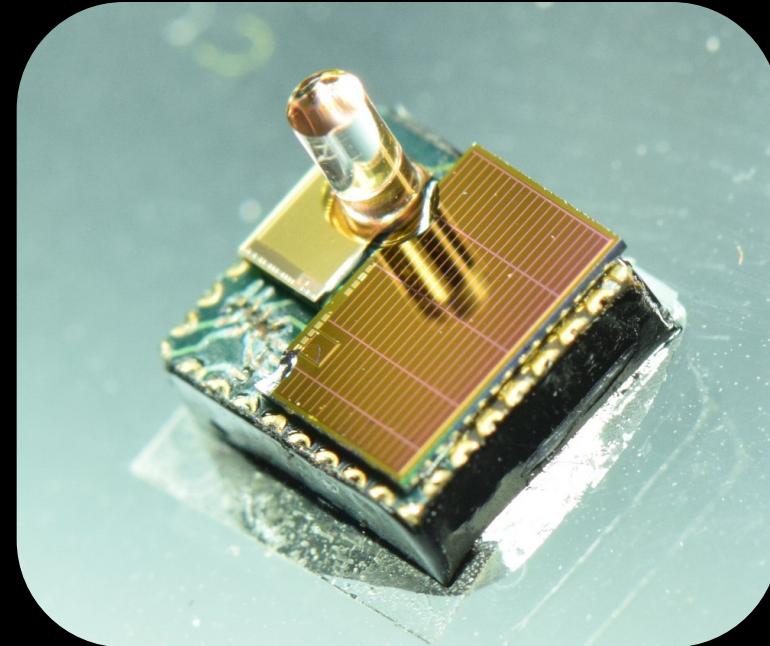
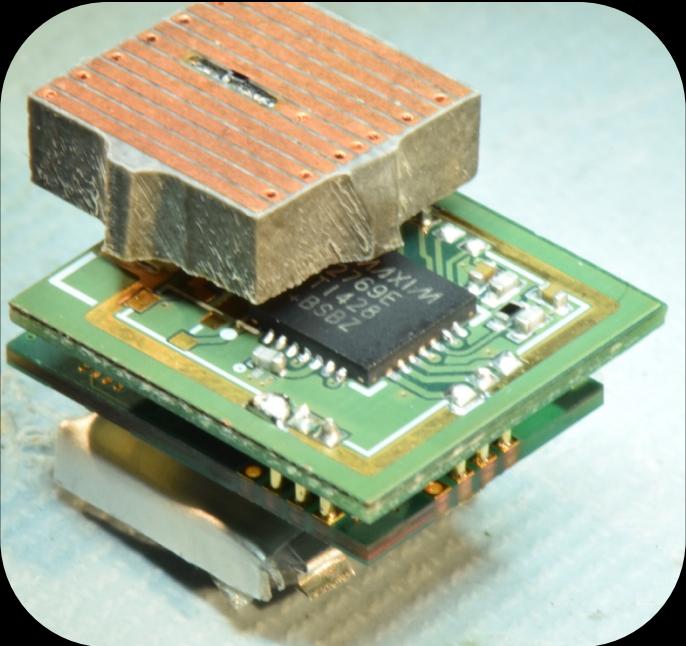
# These primitives enable an architectural shift in system design

- CPU acts as configurator instead of overseer
  - Preprogram temperature sensor to send radio packets
  - Not unlike modern SOCs (sleepwalking,  $\mu$ DMA), but distributed



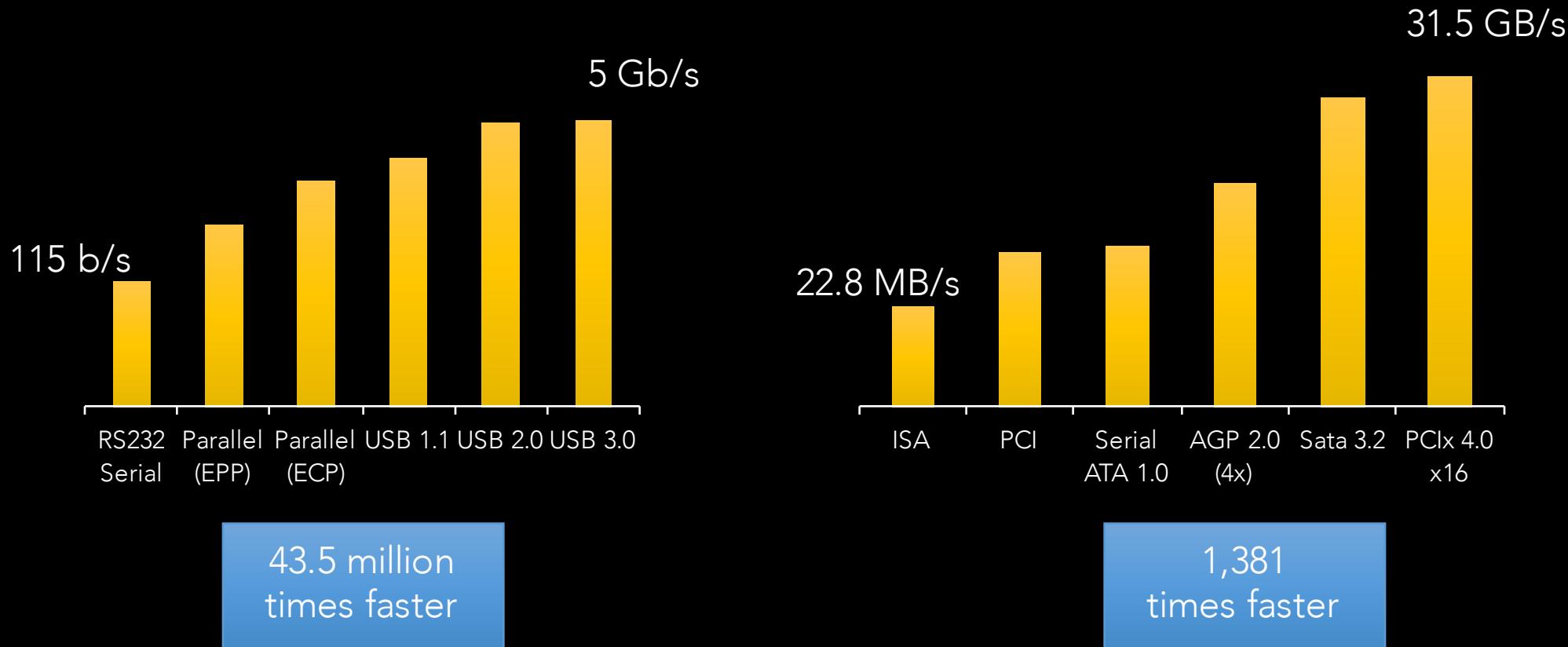
# Seamless and transparent interaction between power-aware and power-oblivious chips

- Facilitates integration with COTS chips

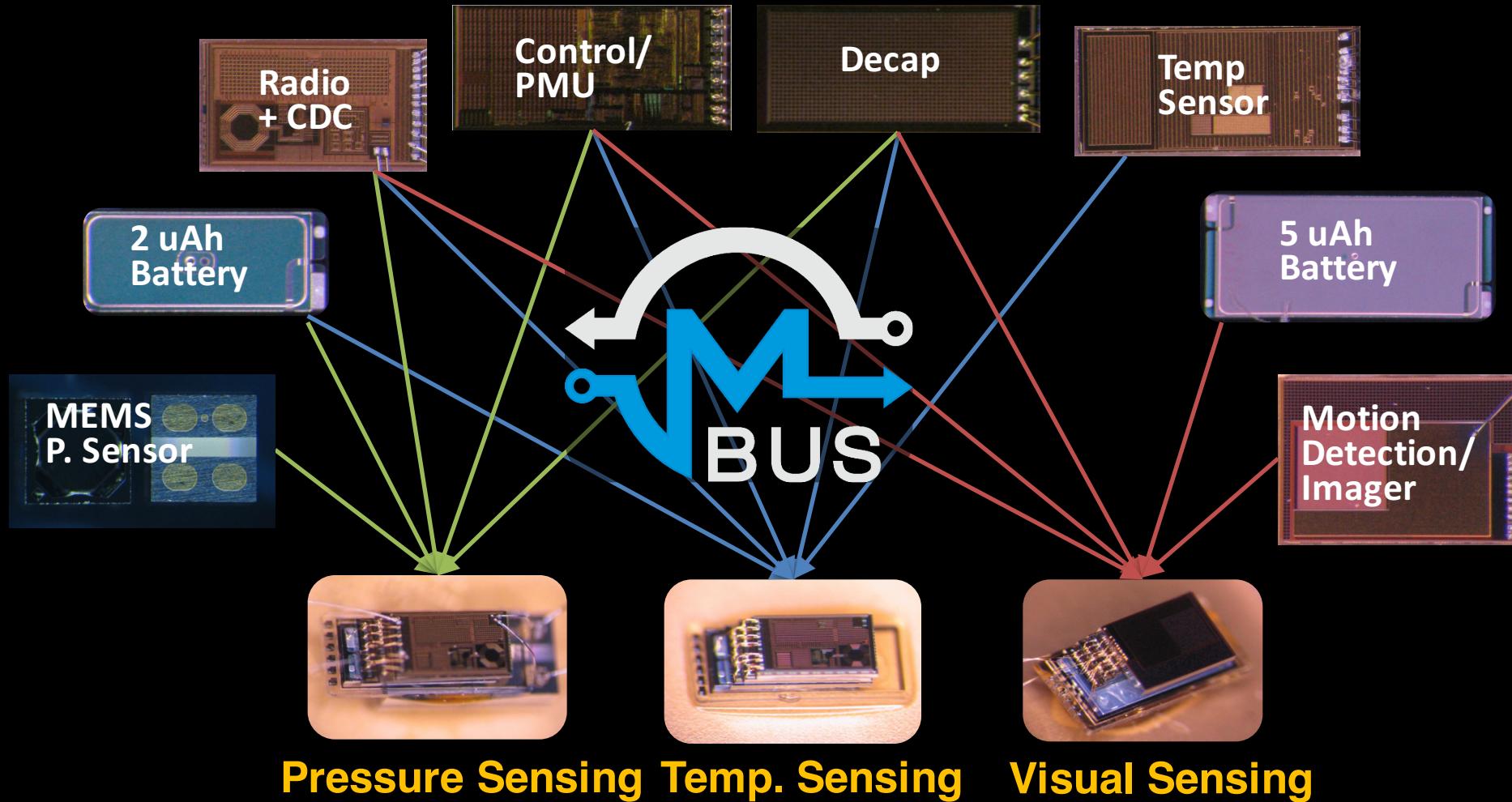


\*No current COTS chip support MBus, these integrations leverage more traditional buses still

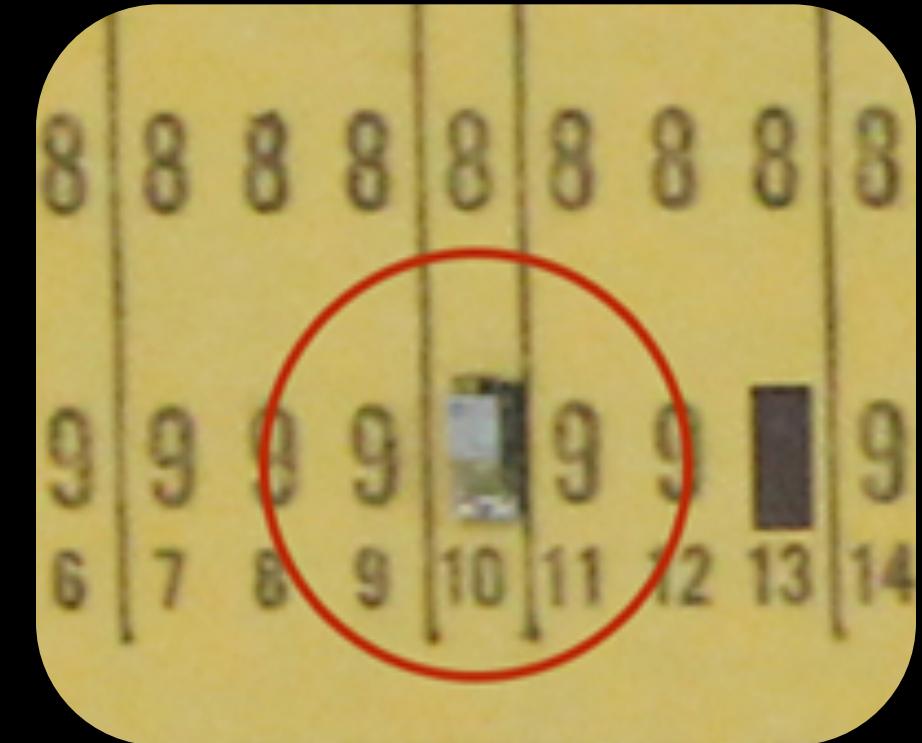
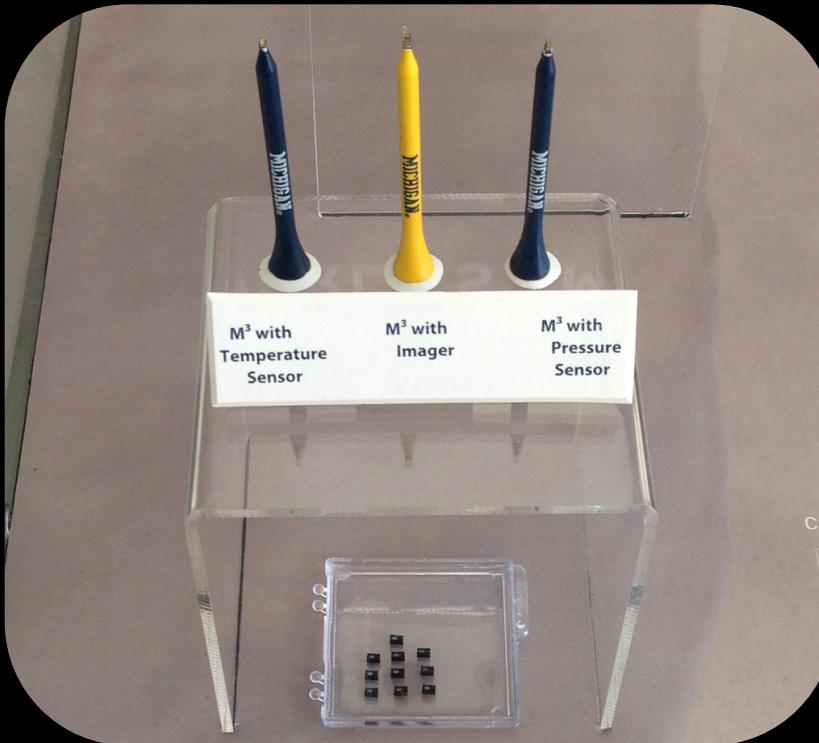
# The majority of interconnect research is focused on performance at the expense of power and area



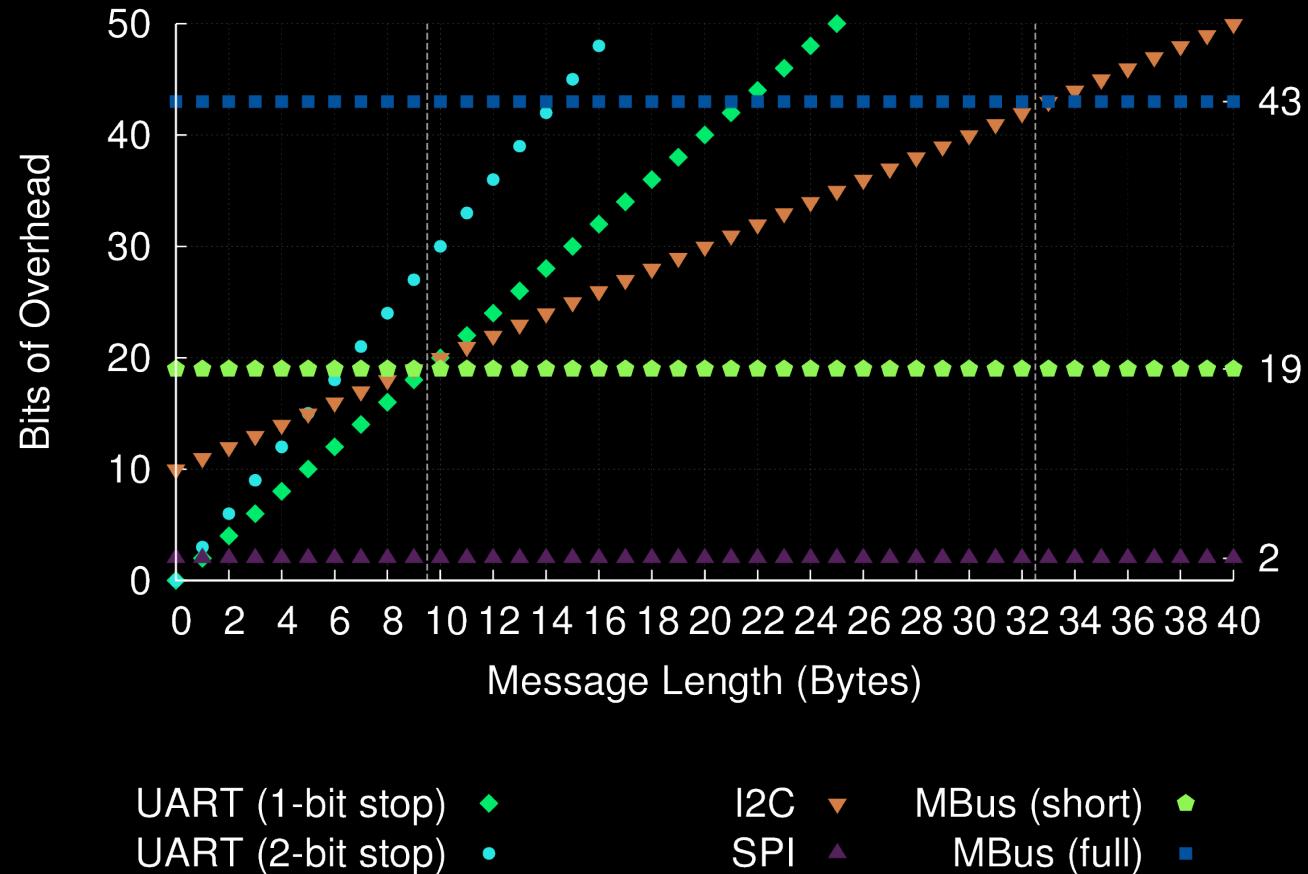
# Specification and Verilog at <http://mbus.io>



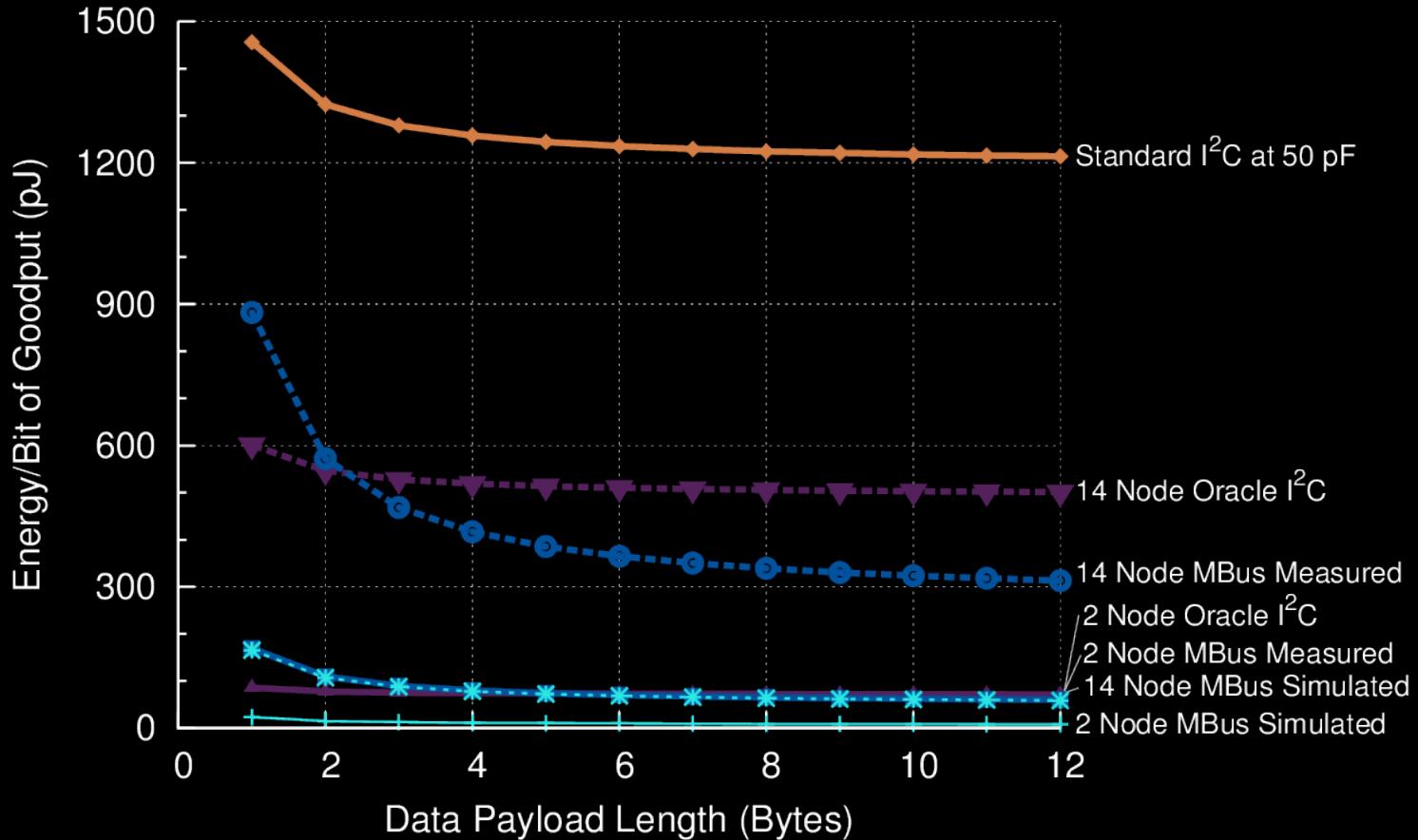
# MBus-based smart dust now on display at the computer history museum



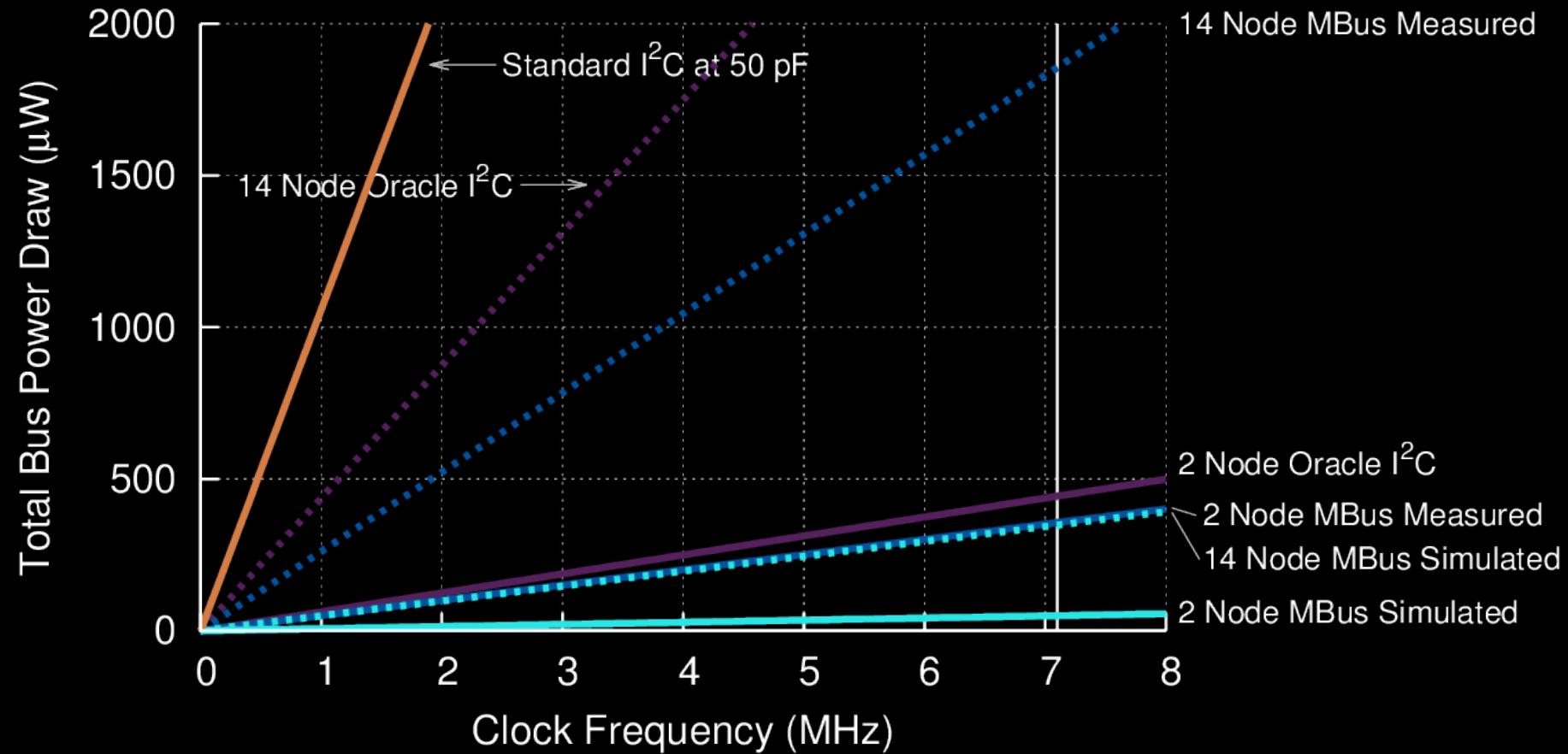
# Protocol Overhead and message length



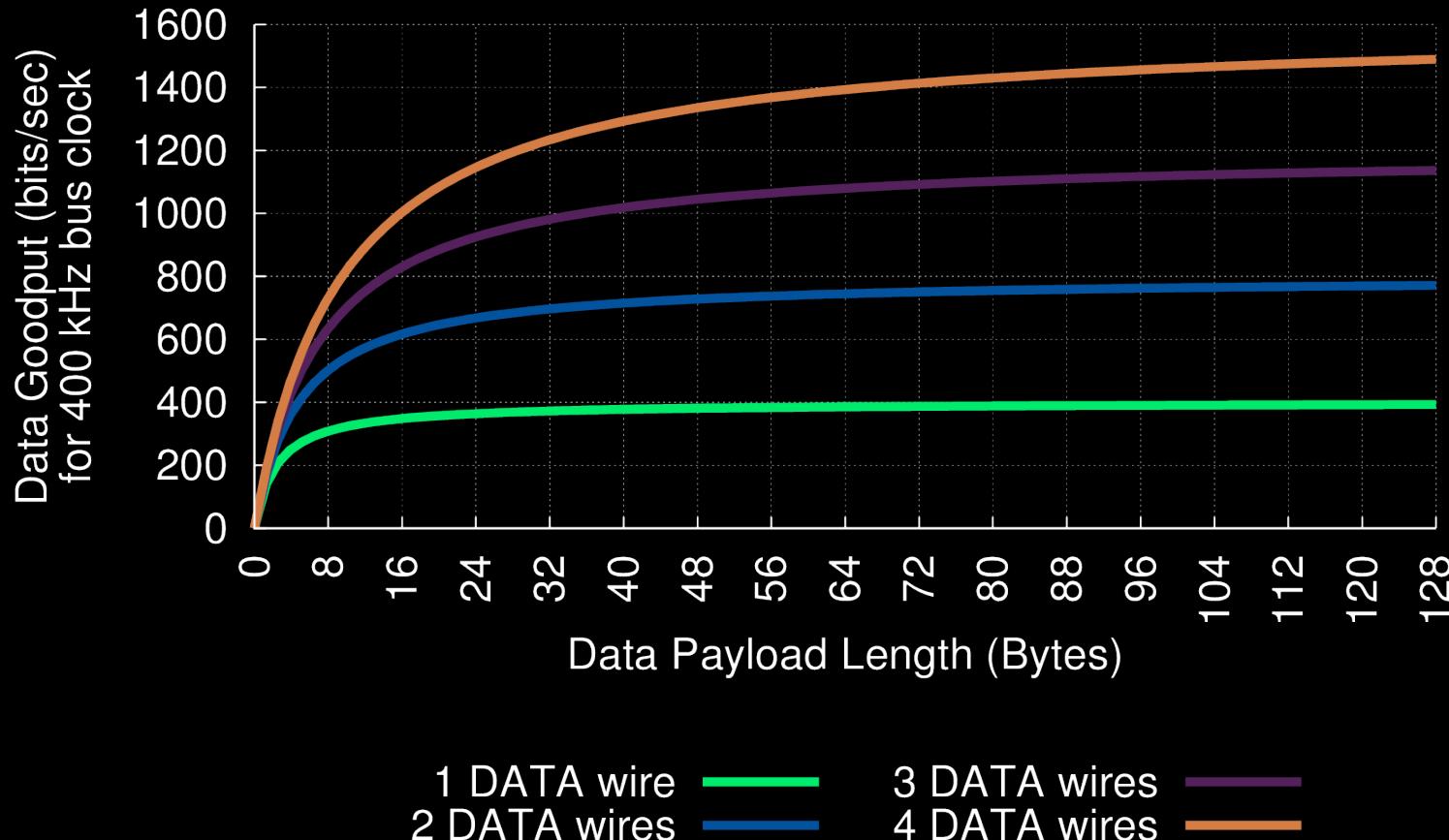
# Energy per bit of goodput (useful data)



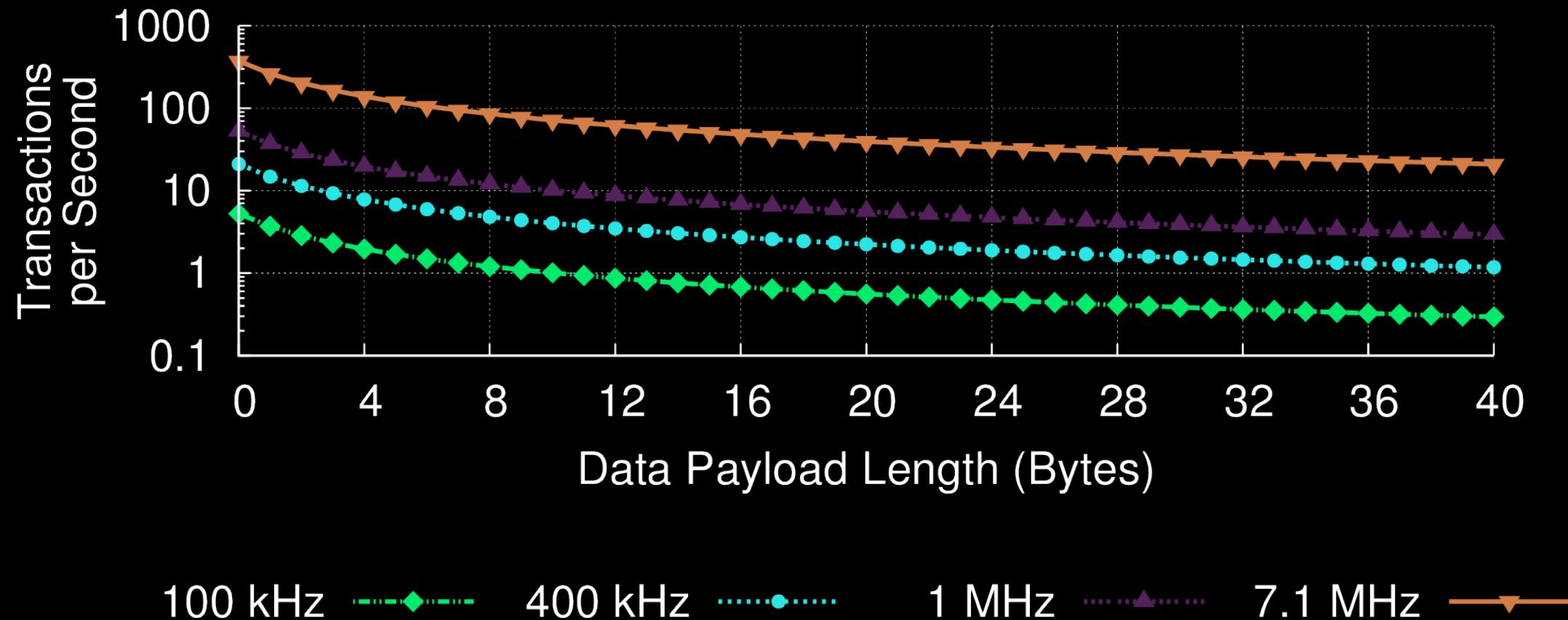
# Power Draw Comparison



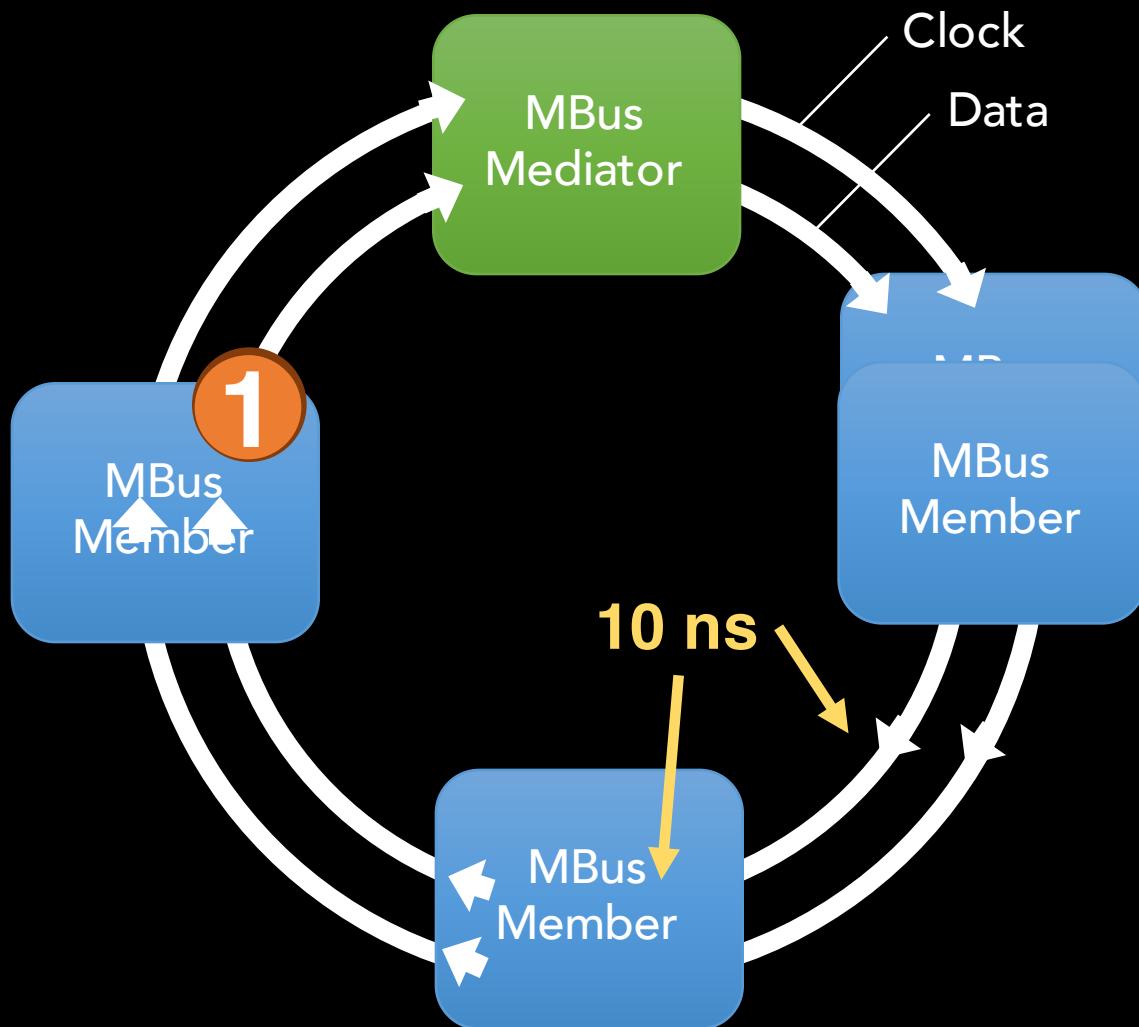
# Goodput of parallel MBus



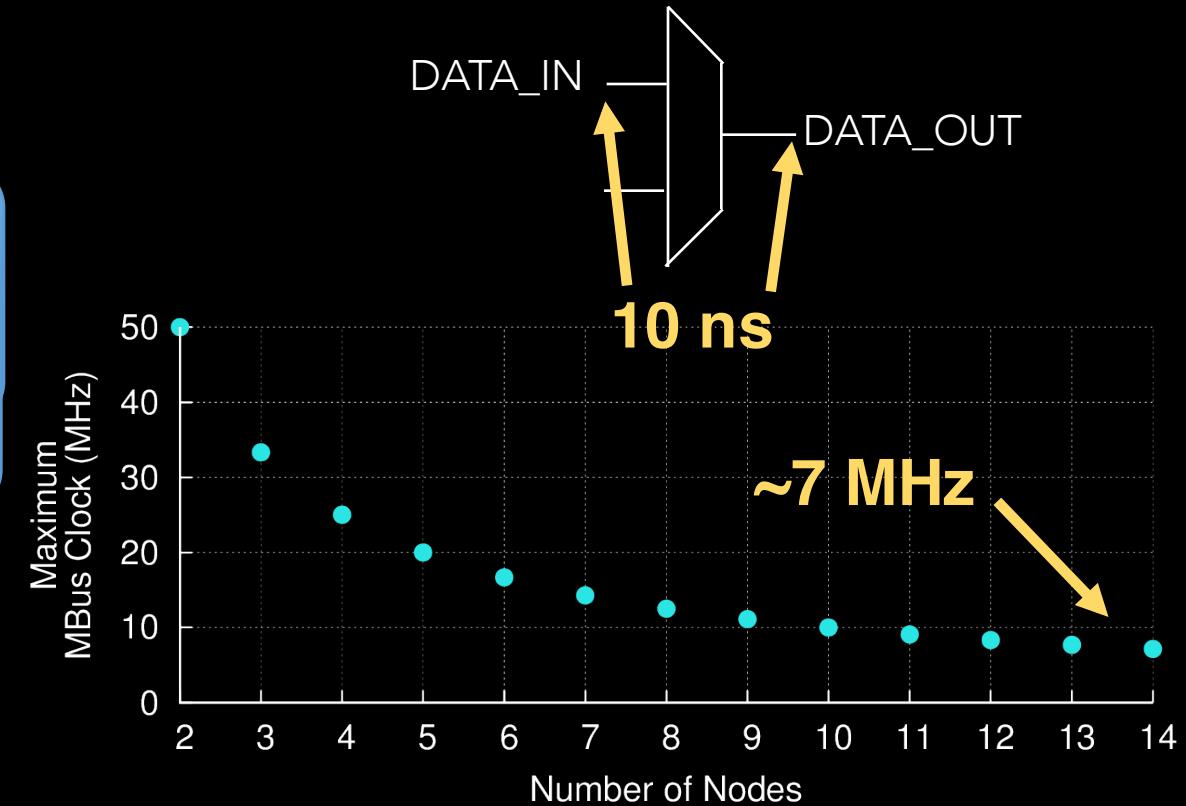
# Saturating Transaction Rate



# Adding additional nodes does not have significant impact on MBus latency

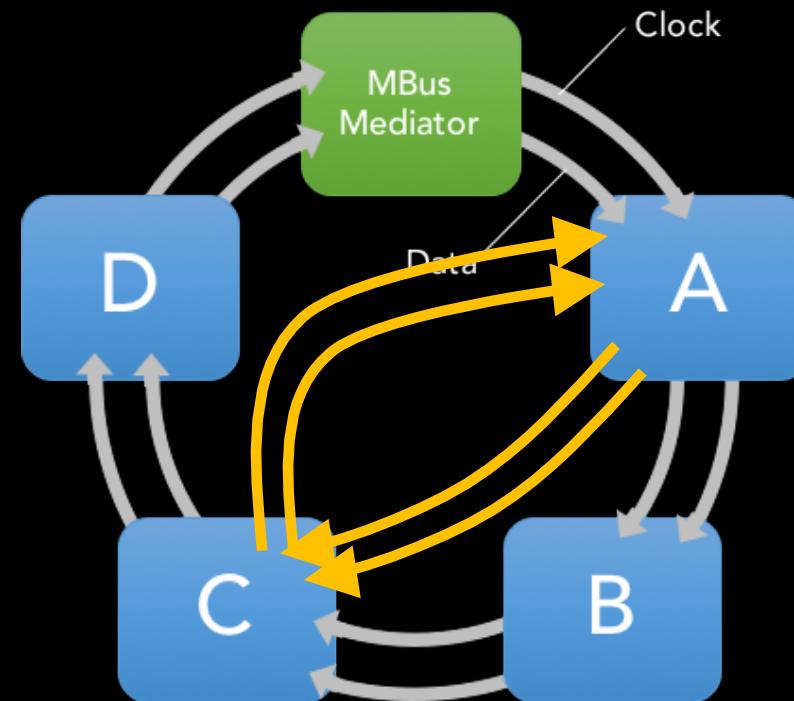
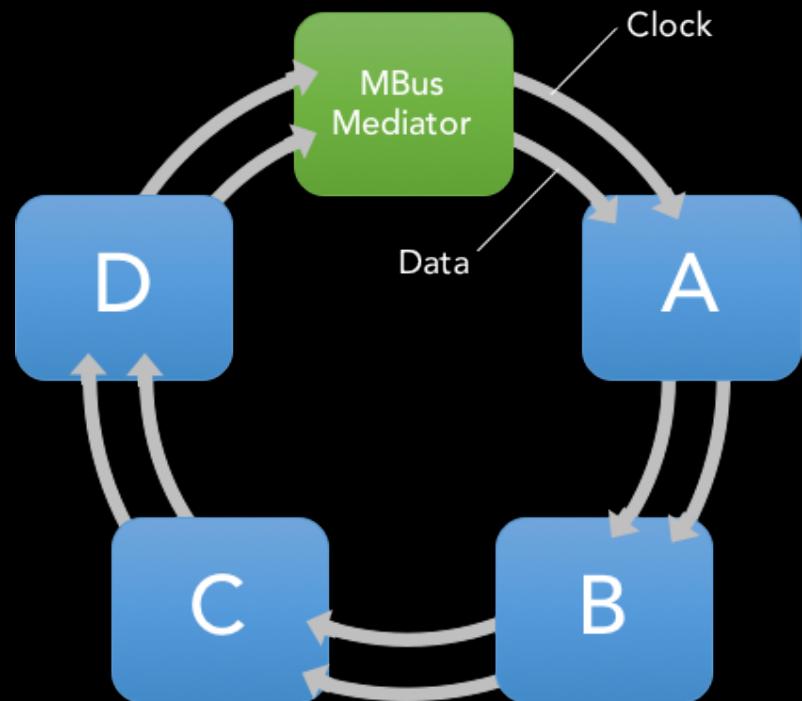


- Recall: “shoot through”

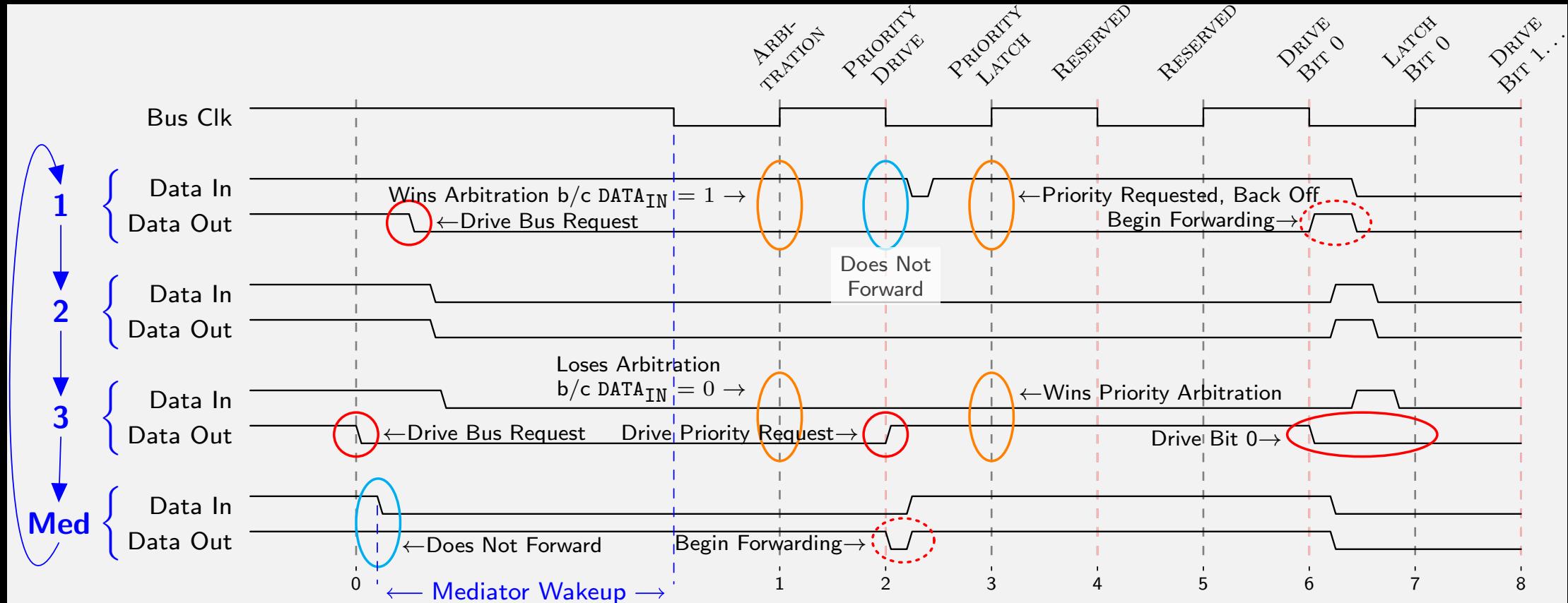


# Some low-hanging fruit...

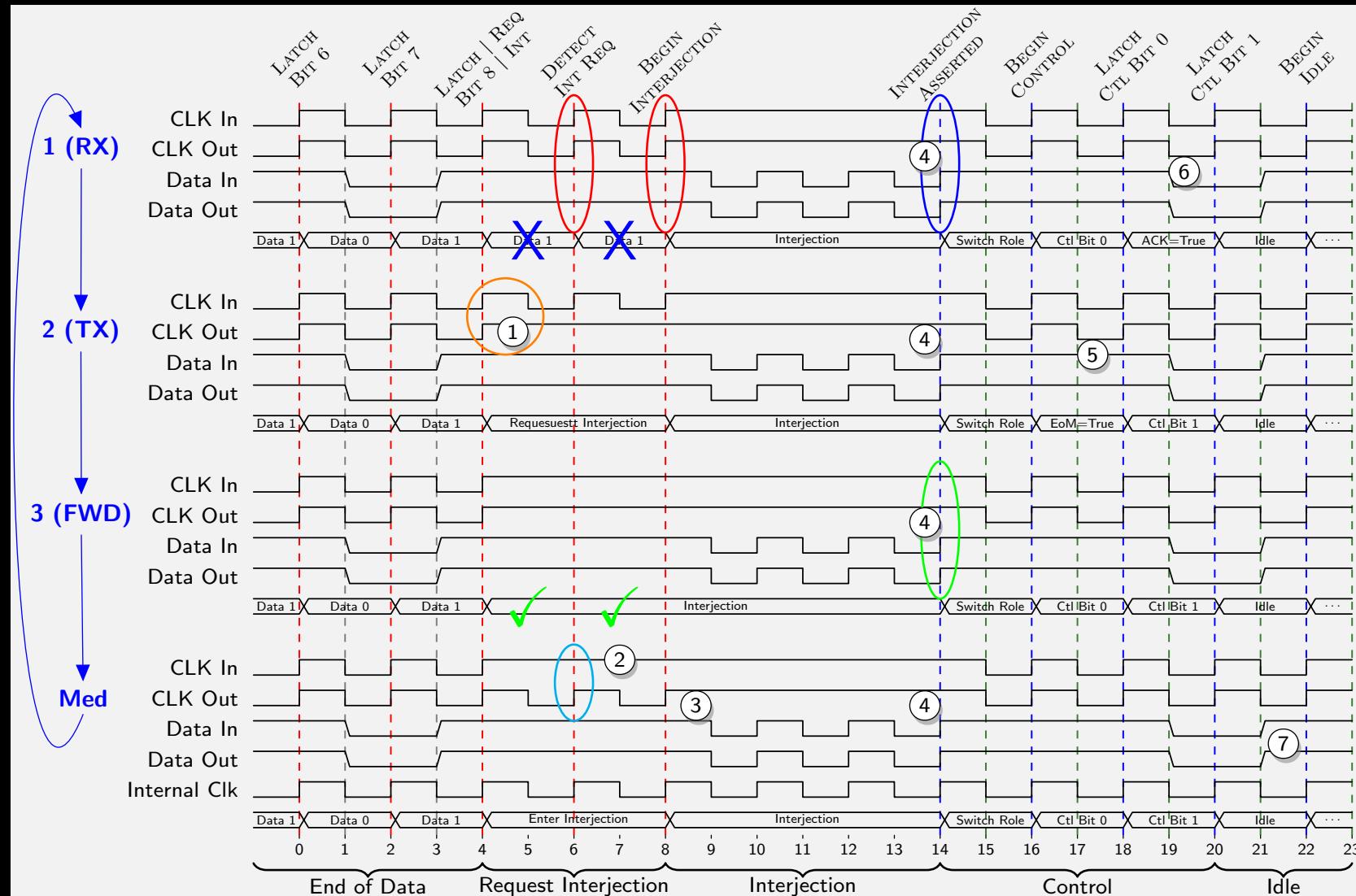
- Full Duplex is trivial
- “Selectively parallel”



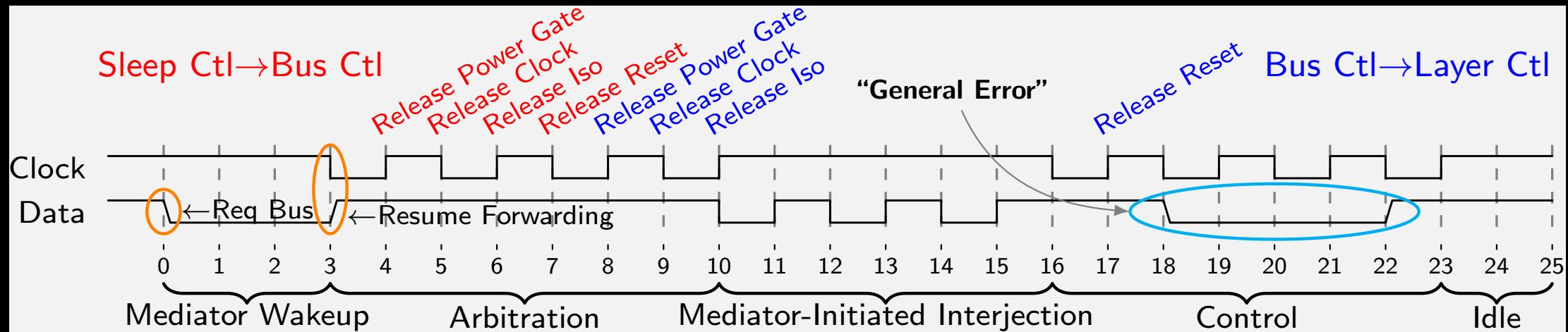
# Arbitration Detail



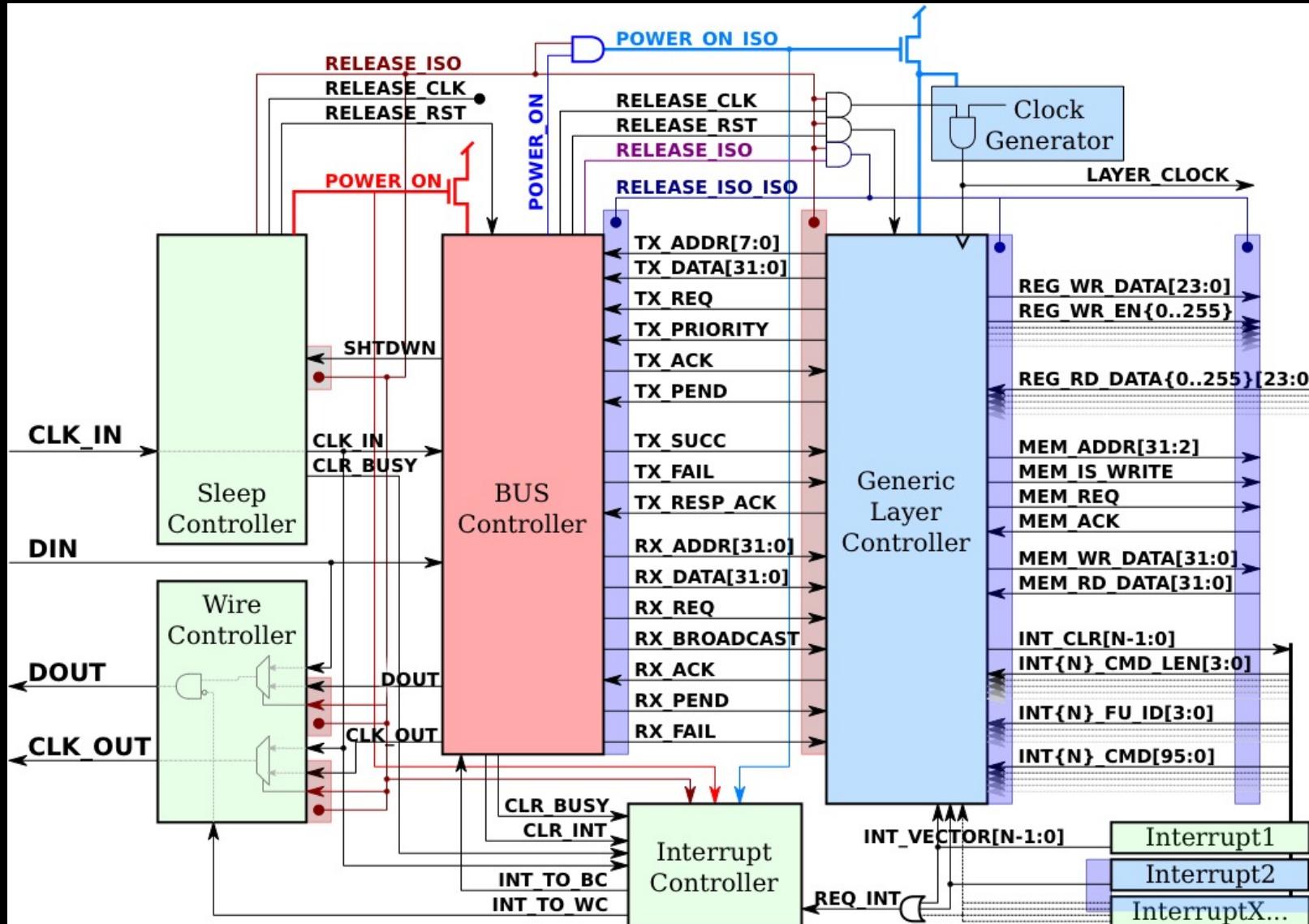
# Interjection Detail



# Tertiary node power-on request



# Hierarchical Power Domains



# Implementation

Module	Verilog SLOC	Gates	Flip-Flops	Area in 180 nm
Bus Controller	947	1314	207	27,376 $\mu\text{m}^2$
<i>Optional</i>				
Sleep Controller	130	25	4	3,150 $\mu\text{m}^2$
Wire Controller	50	7	0	882 $\mu\text{m}^2$
Interrupt Controller	58	21	3	2,646 $\mu\text{m}^2$
Total	1185	1367	214	37,200 $\mu\text{m}^2$ <sup>§</sup>
<i>Other Buses:</i>				
SPI Master <sup>†</sup>	516	1004	229	37,068 $\mu\text{m}^2$
I <sup>2</sup> C <sup>‡</sup>	720	396	153	19,813 $\mu\text{m}^2$
Lee I <sup>2</sup> C [14]	897	908	278	33,703 $\mu\text{m}^2$

<sup>§</sup> Includes a small amount of additional integration overhead area

<sup>†</sup> SPI Master from OpenCores [32] synthesized for our 180 nm process

<sup>‡</sup> I<sup>2</sup>C Master from OpenCores [10] synthesized for our 180 nm process