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 42nd 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

“Smart Dust” should arrive by ~2017
We have a diverse array of mm-scale components

**CPUs**
- V. Ekanayake '04
- B. Warneke '04

**Communication**
- A. Ricci '09
- P. Chu '97
- J. Brown '13
- M. Crepaldi '10

**Timers**
- Y. Lee '13

**ADCs and Sensors**
- Y. Lee '13
- Y.-S. Lin '08

**Power Management**
- N. Sturcken '13
- S. Hanson '09
- M. Scott '03
And a few mm-scale systems...

The 1cc Computer

Smart Dust

Smart Dew

T. Nakagawa '08

B. Warneke '07

Y. Shapira '08
But where is the next class of computing?

1 per Enterprise
1 per Engineer
1 per Professional

1 per Company
1 per Family
1 per person

100 – 1000’s per person

Graphic originally from Dennis Sylvester
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**
MBus enables a modular, composable ecosystem of mm-scale components

From temperature sensor to pressure sensor: 3 months.
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 μW active, 10’s 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

And volume is **shrinking cubically**

10’s μW active, 10’s nW sleep, DC 0.1%
SPI and I\(^2\)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

Key Properties
- 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

- One Two dedicated I/O lines per slave
- Master controls all communication
  - Interrupts must be out-of-band
I²C has fixed I/O requirements and a decentralized architecture

- I²C, invented by Phillips in 1982
  - Any-to-(m)any on one shared bus

- **Key Properties**
  - Fixed wire count (2)
I²C has fixed I/O requirements and a decentralized architecture

- I²C, 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

Send a “1”: 0 J

Send a “0”: 174 pJ

@400 kHz

\[ V_{DD} = 1.2 \text{ V} \]

No setup, no hold

\[ R = 15.5 \text{ k}\Omega \]

(“the best”)

80% \( V_{DD} \) as 1
The problem is the energy costs of running an open-collector.

- **Send a “1”:** 0 J
- **Send a “0”:** 174 pJ
- **SCL Alone:** 70 μW

- **$V_{DD} = 1.2 \text{ V}$**
- **R = 15.5 kΩ** ("the best")
- **80% $V_{DD}$ as 1**
- **No setup, no hold**
- **@400 kHz**

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

- Send a “1”: 0 J
- Send a “0”: 174 pJ
- SCL Alone: 70 μW

@400 kHz

$V_{DD} = 1.2 \text{ V}$

R = 15.5 kΩ

(“the best”)

80% $V_{DD}$ as 1

No setup, no hold
The energy demands of open-collectors make them unsuitable for mm-scale systems.

- Active Energy Budget: 20 μW
- Not an arbitrary number:
  - Volume Target
  - Lifetime Target

SCL Alone: 70 μW

Two batteries

~ 3.1 mm
Can we modify I²C to bring energy costs in line with mm-scale?
Replace the passive pull-up resistor with active circuitry

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

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

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

The Bad: “I²C-like” is not I²C – required FPGA to integrate with COTS chips

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

Ultra-Constrained Sensor Platform Interfacing
Pat Pannuto, Yoonmyung Lee, Benjamin Kempke, Dennis Sylvester, David Blaauw, and Prabal Dutta
IPSN 2012 (Demo)
“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

<table>
<thead>
<tr>
<th>CPU</th>
<th>Radio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Flash</th>
</tr>
</thead>
</table>
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?
  - I²C-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
- I²C 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

![Diagram of Clock and Data signals with ACK and Idle states]
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
  - (MBus Mediator)
  - Nodes appear to be “always on”
  - No need to know or track power 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, µDMA), 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.

<table>
<thead>
<tr>
<th>Interconnect Type</th>
<th>Bandwidth</th>
<th>Times Faster</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS232 Serial</td>
<td>115 b/s</td>
<td></td>
</tr>
<tr>
<td>RS232 Parallel (EPP)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RS232 Parallel (ECP)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USB 1.1</td>
<td>43.5 million times faster</td>
<td></td>
</tr>
<tr>
<td>USB 2.0</td>
<td>1,381 times faster</td>
<td></td>
</tr>
<tr>
<td>USB 3.0</td>
<td></td>
<td>31.5 GB/s</td>
</tr>
<tr>
<td>ISA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Serial ATA 1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AGP 2.0 (4x)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sata 3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCIx 4.0 x16</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5 Gb/s
Specification and Verilog at http://mbus.io
MBus-based smart dust now on display at the computer history museum
Protocol Overhead and message length

![Graph showing protocol overhead and message length](image)

- UART (1-bit stop)
- UART (2-bit stop)
- I2C
- SPI
- MBus (short)
- MBus (full)
Energy per bit of goodput (useful data)
Power Draw Comparison

![Graph showing power draw comparison with different configurations.](image-url)
Goodput of parallel MBus

![Graph showing goodput for parallel MBus with different data payload lengths and number of wires. The y-axis represents data goodput (bits/sec) for 400 kHz bus clock, and the x-axis represents data payload length (Bytes). Different line colors represent 1 DATA wire, 2 DATA wires, 3 DATA wires, and 4 DATA wires.]
Saturating Transaction Rate

[Graph showing the relationship between data payload length (in bytes) and transactions per second, with different data rates (100 kHz, 400 kHz, 1 MHz, 7.1 MHz).]
Adding additional nodes does not have significant impact on MBus latency

- Recall: “shoot through”

DATA_IN

DATA_OUT

10 ns

~7 MHz

Maximum MBus Clock (MHz)

Number of Nodes
Some low-hanging fruit...

• Full Duplex is trivial

• “Selectively parallel”
Arbitration Detail

- Data In
  - Wins Arbitration b/c DATA_in = 1 → Begin Forwarding
  - Does Not Forward

- Data Out
  - Does Not Forward

- Data In
  - Loses Arbitration b/c DATA_in = 0 → Does Not Forward
  - Wins Priority Arbitration

- Data Out
  - Wins Priority Arbitration
  - Drive Bit 0

- Med
  - Mediator Wakeup

- Priority Drive
  - Priority Requested, Back Off

- Drive Priority Latch
  - Reserved

- Drive Priority Reserved

- Drive Bit 0

- Latch Bit 0

- Drive Bit 1...
Interjection Detail

- CLK In
- CLK Out
- Data In
- Data Out
- Data 1
- Data 0
- Internal Clk
- Latch Bit 0
- Latch Bit 1
- Latch Bit 2
- Latch Bit 3
- Latch Bit 4
- Latch Bit 5
- Latch Bit 6
- Latch Bit 7
- Detect Int Req
- Begin Interjection
- Interjection Asserted
- End of Data
- Request Interjection
- Interjection
- Control
- Idle
- Latch Ctrl Bit 0
- Latch Ctrl Bit 1
- Ctl Bit 0
- Ctl Bit 1
- Ctl Bit 2
- Ctl Bit 3
- Ctl Bit 4
- Ctl Bit 5
- Ctl Bit 6
- Ctl Bit 7
- Med

1 (RX)
2 (TX)
3 (FWD)
Tertiary node power-on request
Hierarchical Power Domains
## Implementation

<table>
<thead>
<tr>
<th>Module</th>
<th>Verilog SLOC</th>
<th>Gates</th>
<th>Flip-Flops</th>
<th>Area in 180 nm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus Controller</td>
<td>947</td>
<td>1314</td>
<td>207</td>
<td>27,376 µm²</td>
</tr>
<tr>
<td><strong>Optional</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sleep Controller</td>
<td>130</td>
<td>25</td>
<td>4</td>
<td>3,150 µm²</td>
</tr>
<tr>
<td>Wire Controller</td>
<td>50</td>
<td>7</td>
<td>0</td>
<td>882 µm²</td>
</tr>
<tr>
<td>Interrupt Controller</td>
<td>58</td>
<td>21</td>
<td>3</td>
<td>2,646 µm²</td>
</tr>
<tr>
<td>Total</td>
<td>1185</td>
<td>1367</td>
<td>214</td>
<td>37,200 µm²§</td>
</tr>
<tr>
<td><strong>Other Buses:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPI Master‡</td>
<td>516</td>
<td>1004</td>
<td>229</td>
<td>37,068 µm²</td>
</tr>
<tr>
<td>I²C ‡</td>
<td>720</td>
<td>396</td>
<td>153</td>
<td>19,813 µm²</td>
</tr>
<tr>
<td>Lee I²C [14]</td>
<td>897</td>
<td>908</td>
<td>278</td>
<td>33,703 µm²</td>
</tr>
</tbody>
</table>

§ Includes a small amount of additional integration overhead area

‡ SPI Master from OpenCores [32] synthesized for our 180 nm process

‡ I²C Master from OpenCores [10] synthesized for our 180 nm process