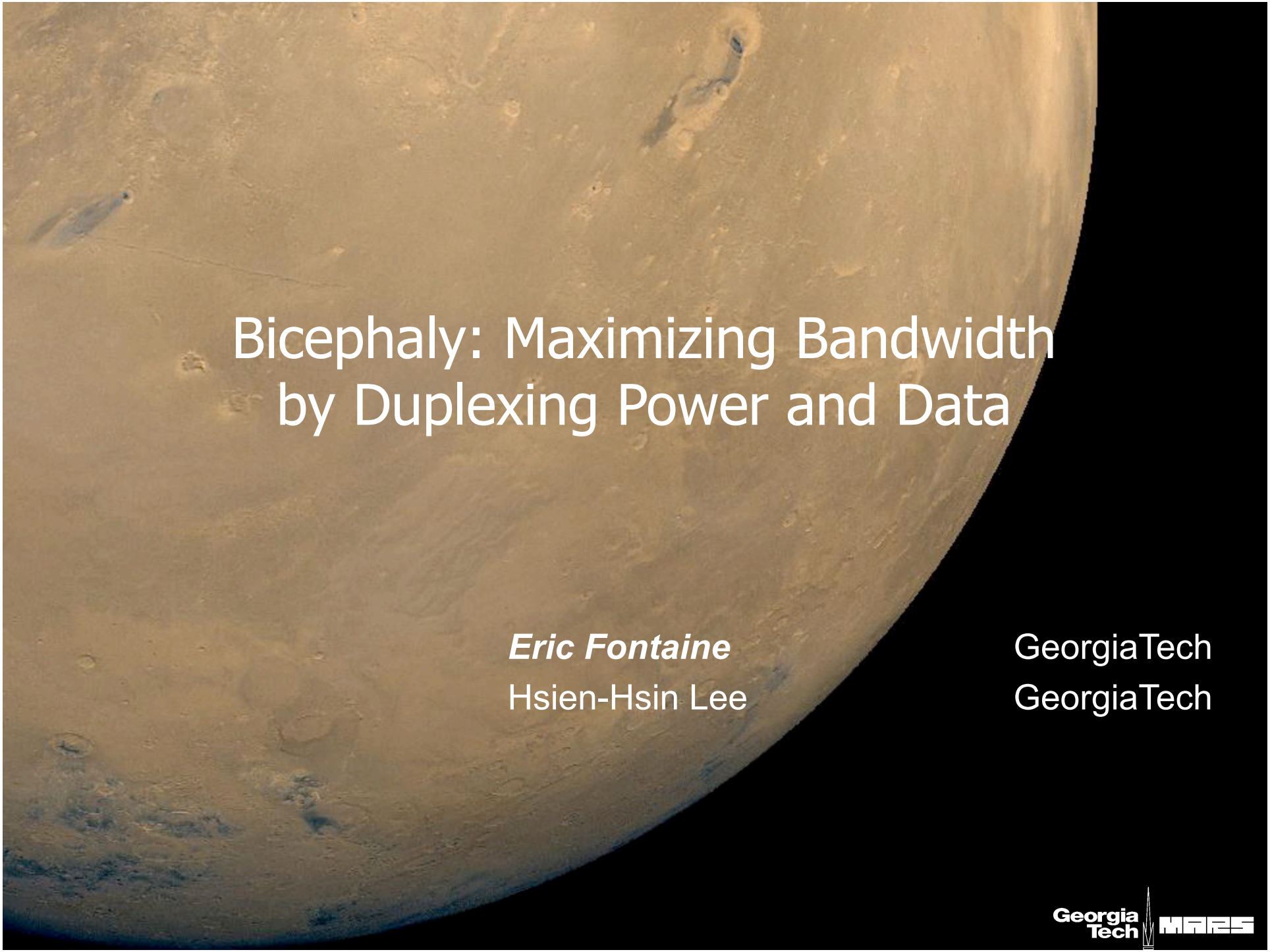


Bicephaly: Maximizing Bandwidth by Duplexing Power and Data

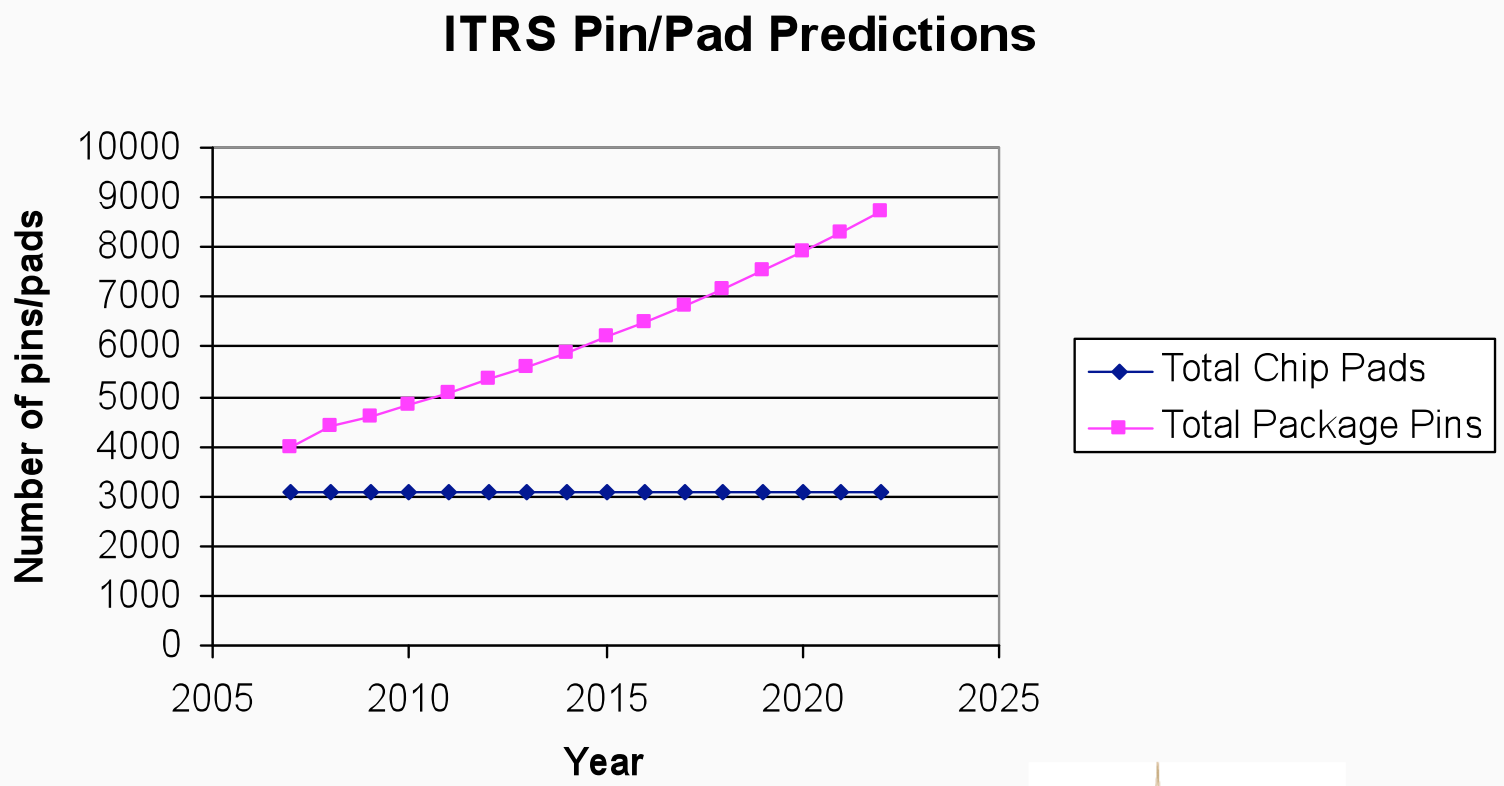
Eric Fontaine GeorgiaTech

Hsien-Hsin Lee GeorgiaTech

Georgia Tech MARS

The Pin Problem

- ITRS predicts slow linear growth in number of pins
  - 2/3 for power and ground, 1/3 for Signal I/O
  - Limited by physical metal properties



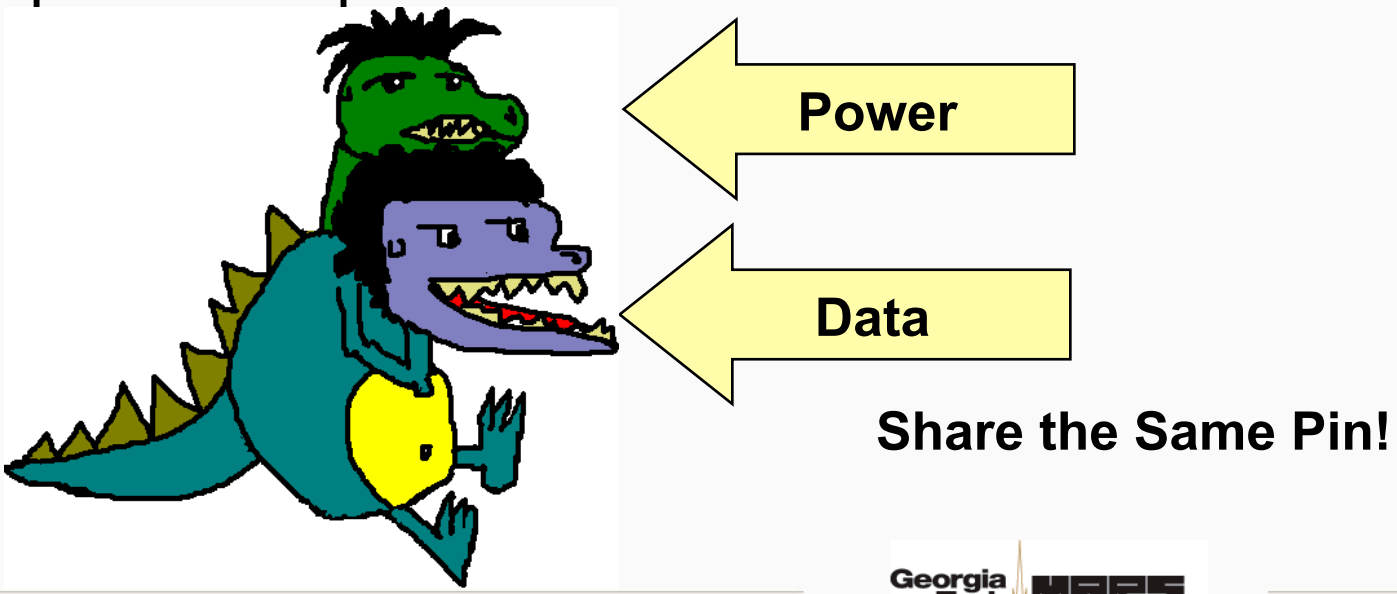
• http://www.itrs.net/Links/2007ITRS/ExecSum2007.pdf

The Bandwidth Problem

- But number cores expected to grow exponentially
  - Greater Power demand
  - Greater Off-chip Bandwidth demand
- How can sustain performance?
- No Data -> NO COMPUTATION
  - Idle cores
- 3-D die-stacked integration only exacerbates
  - Same 2-D real estate for pins
- Bus Frequency scaling and compression has limits

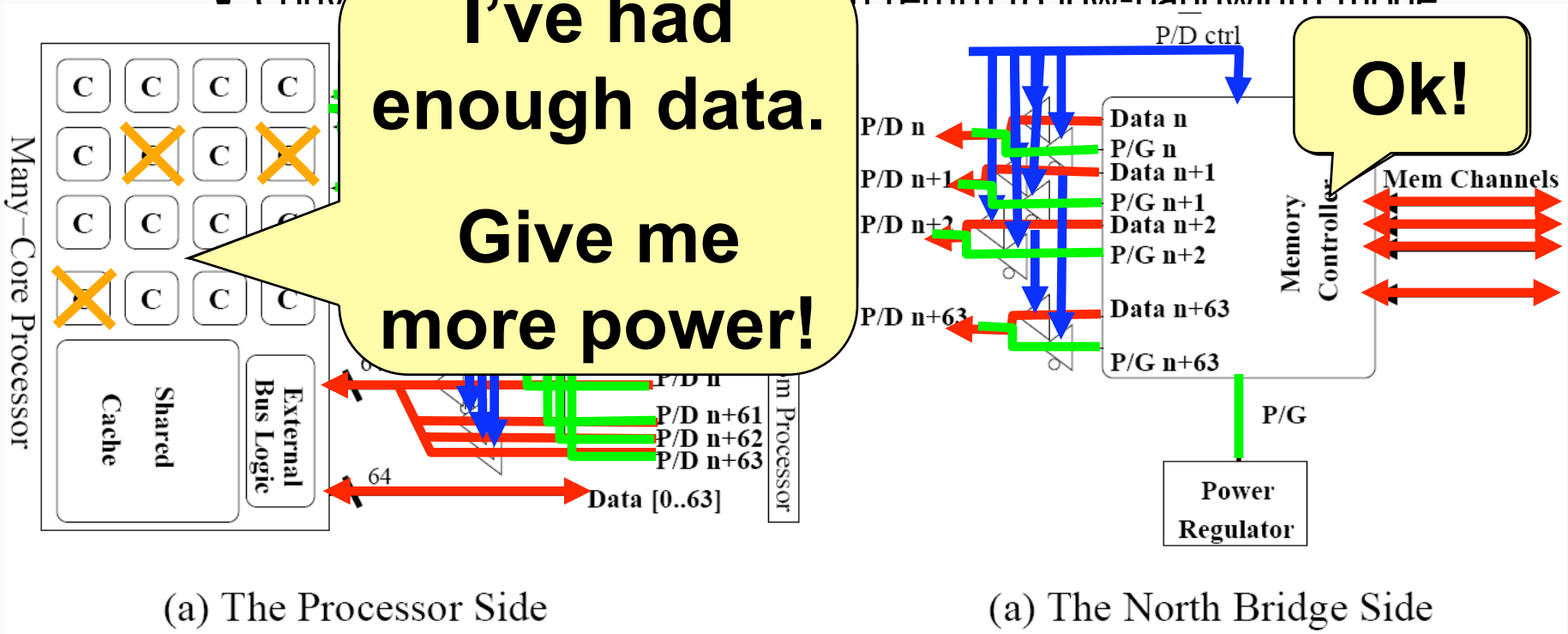
Our Solution: Bicephaly

- Power network designed for worst-case
- But if bandwidth bound, processor does not consume as much power
  - Last level cache miss disrupt data flow
  - Cores/functional units idle waiting for data
- Exploit this fact by dynamically converting power pins into data pins when processor becomes bandwidth bound



How Bicephaly Works

- Processor monitors performance and bus utilization
  - Switch between high-bandwidth and low-bandwidth modes
  - Control signal P/D' ctrl selects power or data lines
  - Duplexable power/data (P/D) lines reconfigured into expanded data bus in high-bandwidth mode
    - Conv… …n return to low-bandwidth mode



Possible Power Saving Techniques

- Disable cores
- Dynamic voltage and frequency scaling of core(s)
- Disable functional units
- Disable cache lines
  - Effective for data-streaming workloads

Physical Challenges

- Bicephaly pins basically use wide t-gates
  - Is full duplex or half duplex better?
- Bus affected by power supply noise
  - Power supply affected by bus noise
- di/dt noise (ground bounce)
- Need decoupling capacitors
  - Capacitors add delay -> slow down bus
- IR drop across power supply network
- Dynamic Reconfiguration Mechanism
  - How long to wait for fluctuations to die down?
  - Stagger disabling?

Floorplaning Challenges

- Which pins to reconfigure?
  - Avoid large local fluctuations in power supply network
    - Distribute reconfigurable pins evenly across chip?
    - Give each core separate power supply network?
  - How synchronize communication?
    - Transfer data across chip needs global pipelined wires
    - Need to synchronize with memory controller

Optimization Challenges

- Control logic to switch modes
  - How often to switch?
    - Does pipeline have to be flushed?
  - Avoid switching too frequently
    - Use upper/lower thresholds
  - Must access performance counters
    - Communicate values across chip
    - What performance counters to use?
      - FSB utilization, IPC, L2 miss rate, # memory accesses,…
    - Must use transistors to evaluate expression
- How reach optimal tradeoff?
  - How many duplex pins to use?
  - Balance data delivery / data consumption

Summary: Maximize performance by duplexing power and data over same pin.

Questions?