Saving Power in the Data Center

Marios Papaefthymiou
Energy Use in Data Centers

• Several MW for large facilities

• Responsible for sizable fraction of US electricity consumption
  – 2006: 1.5% or 61 billion kWh [source: EPA, 2007]
  – 2013: 2.5% or 91 billion kWh [source: NDRC, 2014]
  – 2014: 1.8% or 70 billion kWh [source: LBNL, 2016]
Where Does All This Energy Go?

- Power Usage Effectiveness (PUE) = \( \frac{\text{IT} + \text{Overhead}}{\text{IT}} \)
- Global average \( \sim 1.7 \) [source: Uptime Institute 2014 Data Center Survey]
- Industry leaders pushing PUE below 1.1
IT Energy

• Server
  – Maximum power consumption
  – Server utilization (pushing 50% in hyperscale data centers)
  – Ability to scale server power with utilization (power proportionality)
  – > 80% of IT energy

• Storage
  – SSD at 6W/disk
  – Increases in HDD efficiency to continue, approaching SSD
  – Energy per year projected <10 billion kWh beyond 2020

• Network
  – Increases in W/port to continue
  – Energy per year projected <2 billion kWh beyond 2020
Energy Consumption Y2K: No Problem

- Manageable current supply requirements (10s of Amps)
- Thermal limits safely away
  - Chips not melting
- Cloud / mobile revolution still at its infancy
  - Limited motivation to lower energy requirements
- “Voltage scaling” kept power requirements under check
Performance Scaling
1990 – 2000

Performance (GOPS)
Voltage Scaling
1990 – 2000

Supply Voltage (V)

1.5um
1um
0.75um
0.5um
0.25um

1990 1995 2000
Energy Consumption $= f\ (\text{Voltage})$

Constant supply voltage $V$
Energy Consumption = f (Voltage)

\[ \frac{1}{2} CV^2 \text{ consumed on device resistance} \]

\[ \frac{1}{2} CV^2 \text{ stored in capacitive load } C \]

Constant supply voltage \( V \)
Energy Consumption = f (Voltage)

Total energy consumed = $\frac{1}{2} CV^2 + \frac{1}{2} CV^2 = CV^2$
Voltage Scaling
Post-2000

![Graph showing voltage scaling from 1990 to 2010. Supply Voltage (V) decreases significantly over time.]
Performance Scaling
Post-2000

Chips reached their thermal limits
Now What?

• 2000-present: Let’s turn off whatever is not used
  o Clock gating
  o Power gating
  o Multi-core architectures
  o Software-controlled power management

• Limited ability to scale down server power with utilization: Deeper power down \(\rightarrow\) longer idle-to-active \(\rightarrow\) delayed response times

• At the end of the day, we are basically stuck!
Fundamental Physical Limitation

Total energy consumed $= CV^2$
Let’s Modify the Voltage Supply

Gradually transitioning supply voltage

Load $C$

$V$

time
Energy Consumption for Gradual Charging

\[(RC/T) CV^2 \text{ consumed on R of device}\]

Gradually transitioning supply voltage

\[\frac{1}{2} CV^2 \text{ stored in load C}\]
Energy Consumption for Gradual Discharging

Gradually transitioning supply voltage

\[(RC/T) CV^2 \text{ consumed on R of device}\]

\(\frac{1}{2} CV^2 \text{ stored in C returns to supply}\)

Graph of supply voltage vs time:

- Voltage \(V\) increases linearly over time \(2T\)
Putting it All Together

Total energy consumed to charge/discharge
\[ \sim = 2 \left( \frac{RC}{T} \right) CV^2 \]

Gradual recycling of charge saves energy
(assuming $RC/T < 2$)

In fact, the slower the better
Intrinsic Energy Requirements & Reversibility of Computation

- Fundamental question: What are the intrinsic energy requirements of computation?

- Main result (Landauer, 1961): Energy requirements can asymptotically approach zero, provided the computation is reversible (i.e., retains all information required to reproduce the initial state of the computing machine)

- Argument based on statistical thermodynamics: Loss of information $\rightarrow$ change in system entropy $\rightarrow$ energy converted to heat (i.e., wasted)
Reversible Turing Machines and Universal Gates

• Logically reversible Turing Machines (Bennett, 1973)
  o Compute just like a conventional TM, but keep intermediate results
  o Output final result
  o Reverse operation, disposing of intermediate results and returning the machine to its original state

• Reversible universal gates (Fredkin, Toffoli, 1982)
  o Embed function into larger space to yield 1-1 mapping between inputs and outputs

• Feynman Lectures on Computation, T. Hey and R. Allen eds., 1996
A Universal Reversible Gate

A’ = A

If A = 1, then B’ = B and C’ = C

If A = 0, then B’ = C and C’ = B
Early Prototypes (‘90s)

• Reversible pipelines + split-level charge-recovery logic: Pendulum computer (Knight et al., 1995-2000)

• Variety of irreversible charge-recovery circuit families

• AC computers (Athas et al., 1995-2000)
  o Focus on simple reversible function

• Gradual charge/discharge through inductors or capacitor “ladders”

• Issues
  o High overheads: Too many bits/gates
  o High complexity: Too many wires
  o Slow operation
Pop Quiz

Give the simplest interesting reversible function.
(“interesting” is defined as “used in every digital system and consumes a lot of power.”)
Pop Quiz

Give the simplest interesting reversible function. ("interesting" is defined as "used in every digital system and consumes a lot of power.")

\[ F(x) = x \]

"Identity function"

a.k.a.

"clock" (in synchronous digital systems)
Chip Designs with Resonant Clock Mesh Distribution Networks

- GHz clock CICC’06
- IBM Cell BE ISSCC ‘09
- 2GHz FPU A-SSCC ‘10
- IBM Power8 ISSCC ‘14
- IBM System z ISSCC ‘15
- AMD Piledriver ISSCC ‘12
- AMD Steamroller ISSCC ‘15

- DWT ASIC ISLPED’03
- GHz FIR VLSI’07 CICC’07
- ARM 926EJ-S ESSCIRC ‘09

- 200MHz 2015
- 250nm 5+ GHz
- area < 1mm² 22nm
- silicon prototypes area > 20mm² high-volume commercial CPUs
Clock Mesh

Typical solution for multi-GHz processors

Pros
- All-metal mesh shorts clock buffer outputs → very low skews
- Mesh isolates local timing → simplifies late-design ECOs

Cons
- Large capacitance of clock mesh leads to high power consumption
Resonant Clock Mesh

*Provides all the performance benefits of a clock mesh AND reduces power*

**Pros**
- All-metal mesh shorts clock buffer outputs → very low skews
- Mesh isolates local timing → simplifies late-design ECOs

**Cons**
- Large capacitance of clock mesh leads to high power consumption
4+ GHz x86 “Piledriver” Core

- High-volume 64-bit x86 core in 32nm, 11 metal layers
- 100 on-chip inductors, 0.7nH to 1.2nH
- Up to 30% clock power savings
- 5% to 10% reduction in total chip power, depending on workload profile

Sathe et al., ISSCC ‘12
Simplified Model of Dual-Mode Clock Network
Global Clock Organization and Distribution

- Large variation of capacitance across the chip
- Palette of 5 inductors (0.7nH to 1.2nH)
- Integer linear program to select inductor values and size clock wires to minimize clock skew
Inductor Design

~100 um
Simulated Clock Waveforms
Clock Power Savings vs Frequency

- Efficiency: Percentage clock power savings over cclk
- rclk_square_x → Clock driver strength modulation of x/7
Resonant Mega-Mesh for IBM z13™

- 545mm² (80% of chip area), encompassing 8 cores and L3
- Single mesh enables 2x clock frequency in L3 → reduced memory latency
- 22nm high-k CMOS SOI, 17 metal layers
- Operating range: 4.5GHz to 5.5GHz
- 50% savings in final-stage clock mesh power
- 8% savings in total chip power

Shan et al., VLSI Symp 2015
Total energy consumed to charge/discharge

\[ \approx 2 \left( \frac{RC}{T} \right) CV^2 \]

**How About Logic?**

Gradual recycling of charge saves energy
*(assuming RC/T < 2)*

*In fact, the slower the better*

If operation is too slow, then result is not interesting.

True for all-metal clock networks, but is it true for logic?
13.8μW Binaural Dual-Microphone ANSI S1.11 Filter Bank for Hearing Aids

- 10x reduction in energy consumption per input in 65nm over published state of the art in 40nm

Wu et al., ISSCC 2017
Conclusion

- Energy-recycling can save significant amounts of power in high-end server chips (5% to 10% of total chip power) without sacrificing performance
- Successfully deployed in clock distribution networks of high-volume multi-GHz server processors (AMD, IBM)
- Significant potential for reducing energy consumption in logic
  - Labor intensive custom logic design
  - Increased latency due to micropipelining
  - Large capacitors require large inductors
Acknowledgments

• Alex Ishii, Cyclos co-founder

• My U-M students:
  – Suhwan Kim (Seoul National U.)
  – Conrad Ziesler (Apple)
  – Joohee Kim (Cyclos)
  – Juang-Ying Chueh (TSMC)
  – Visvesh Sathe (U. Washington)
  – Jerry Kao (GlobalFoundries)
  – Wei-Hsiang Ma (Intel)
  – Tai-Chuan Ou (Apple)

• ARO, DARPA, NSF