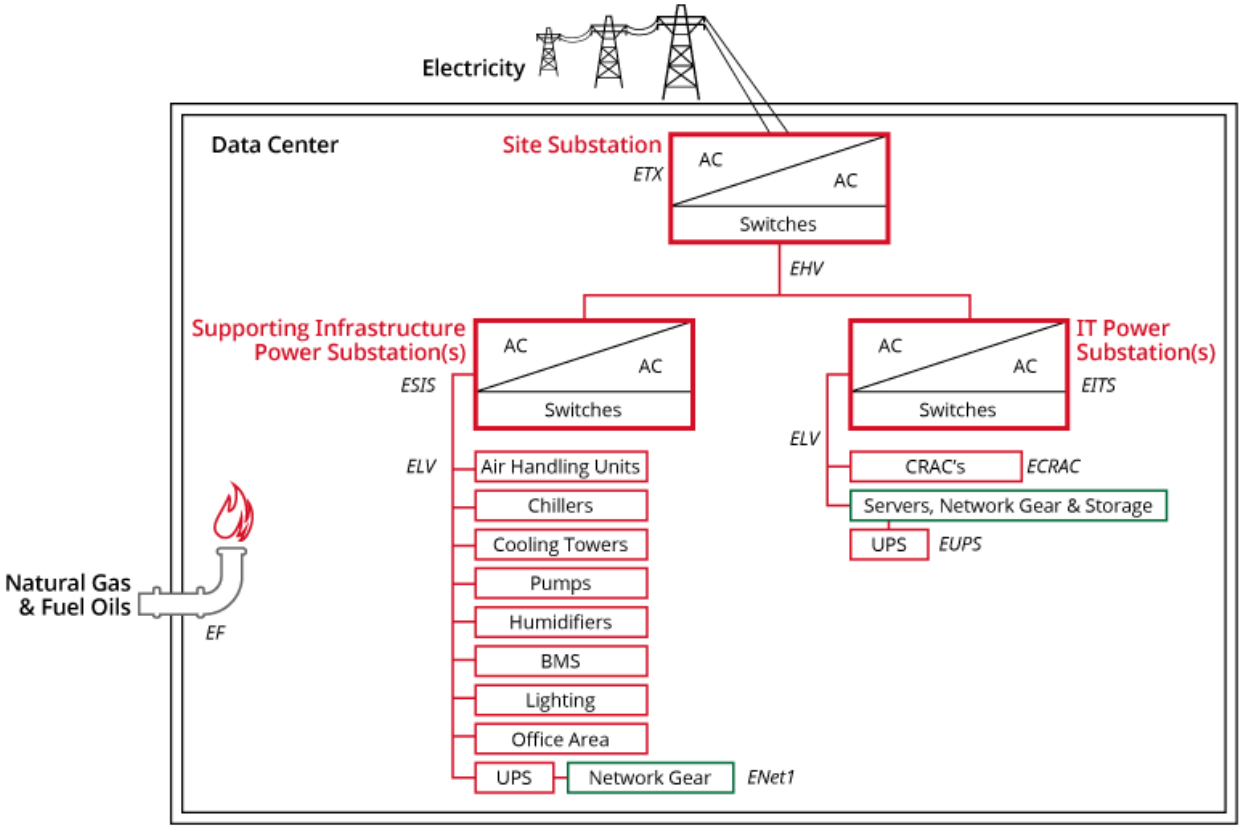
Saving Power in the Data Center

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?



Source: Google

- Power Usage Effectiveness (PUE) = (IT + Overhead) / IT
- Global average ~1.7 [source: Uptime Institute 2014 Data Center Survey]
- Industry leaders pushing PUE below 1.1

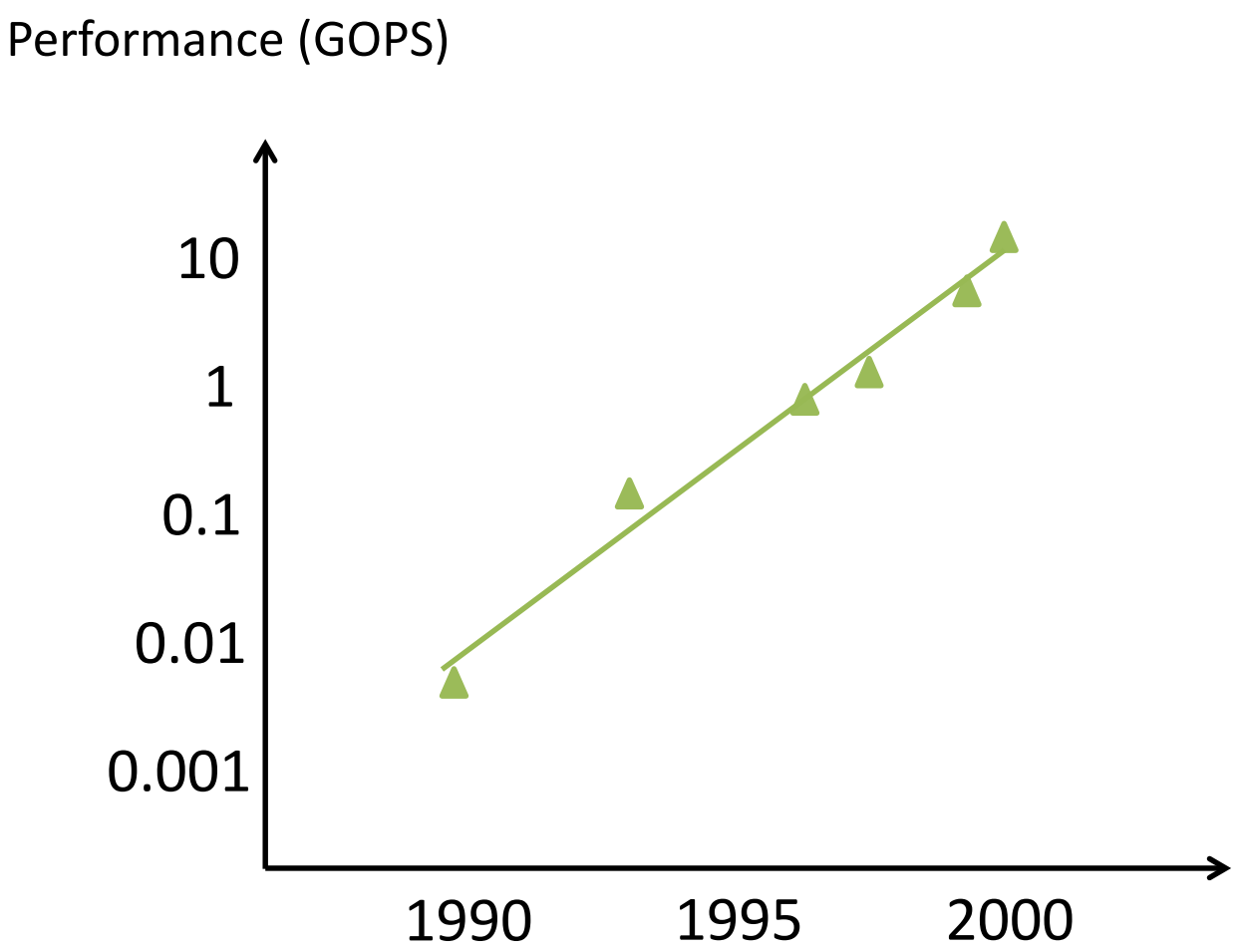
IT Energy

- 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

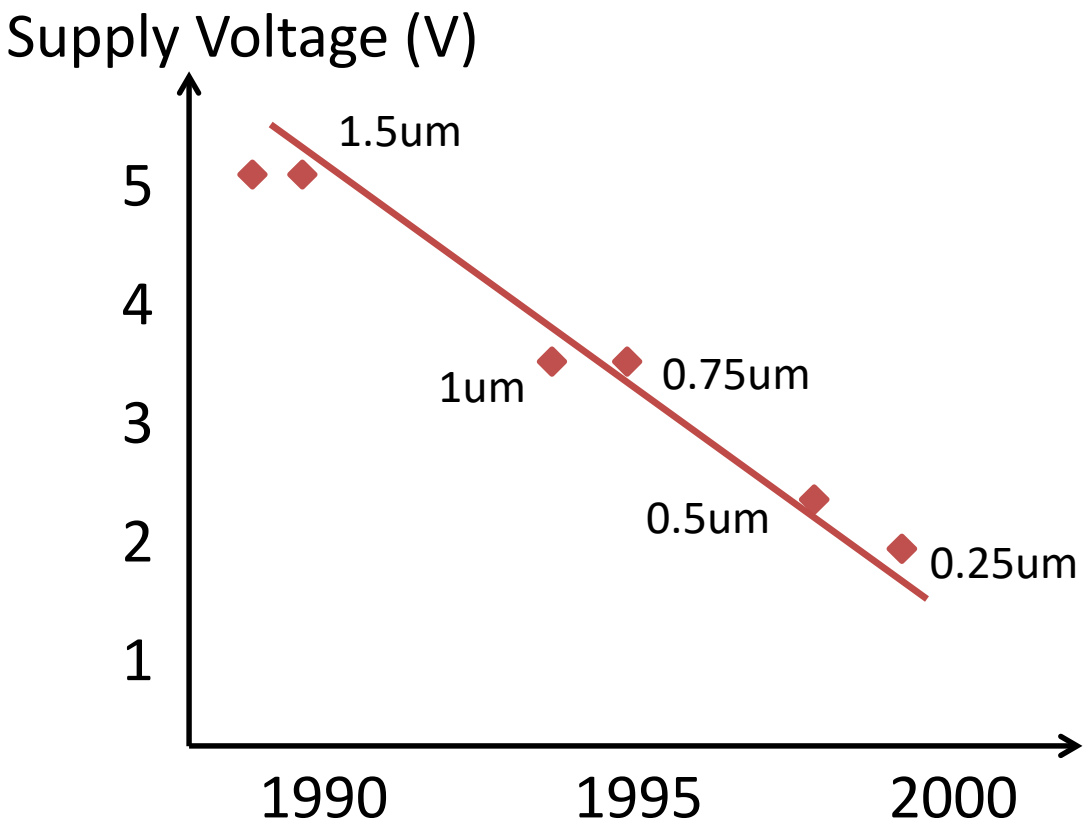
- Manageable current supply requirements (10s of Amps)
- Thermal limits safely away
  - Chips not melting
- Cloud / mobile revolution still at its infancy
  - Limited motivation to lower energy requirements
- “Voltage scaling” kept power requirements under check

Performance Scaling
1990 – 2000



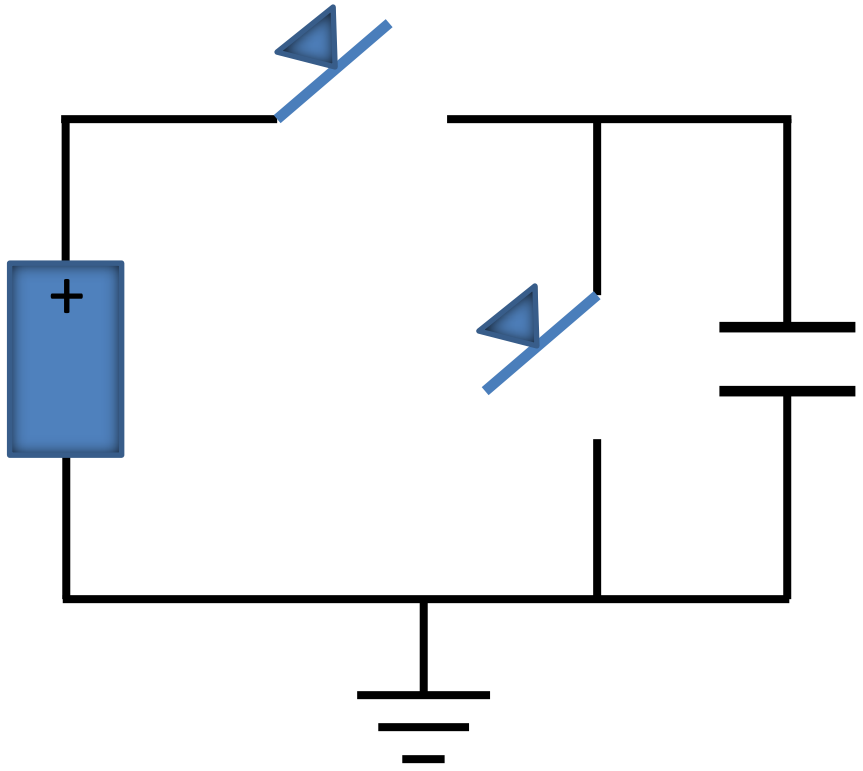
Voltage Scaling

1990 – 2000

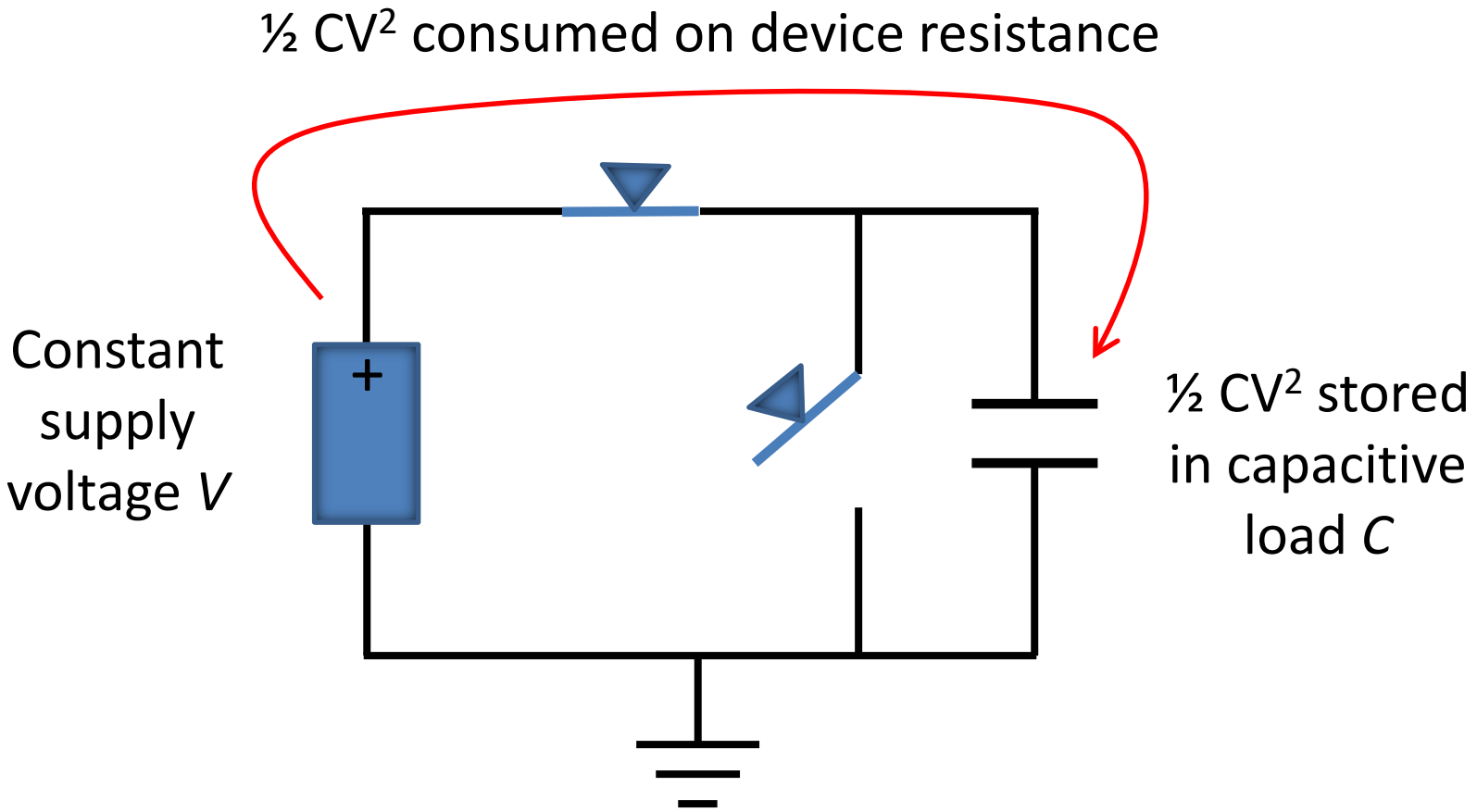


Energy Consumption = f (Voltage)

Constant supply voltage *V*

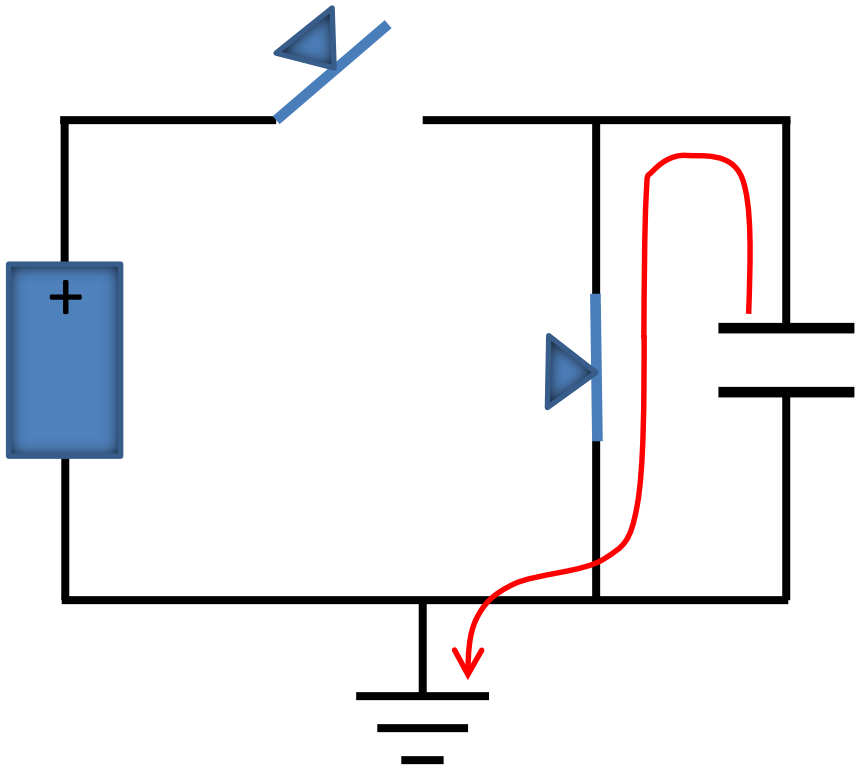


Energy Consumption = f (Voltage)



Energy Consumption = f (Voltage)

Constant supply voltage *V*

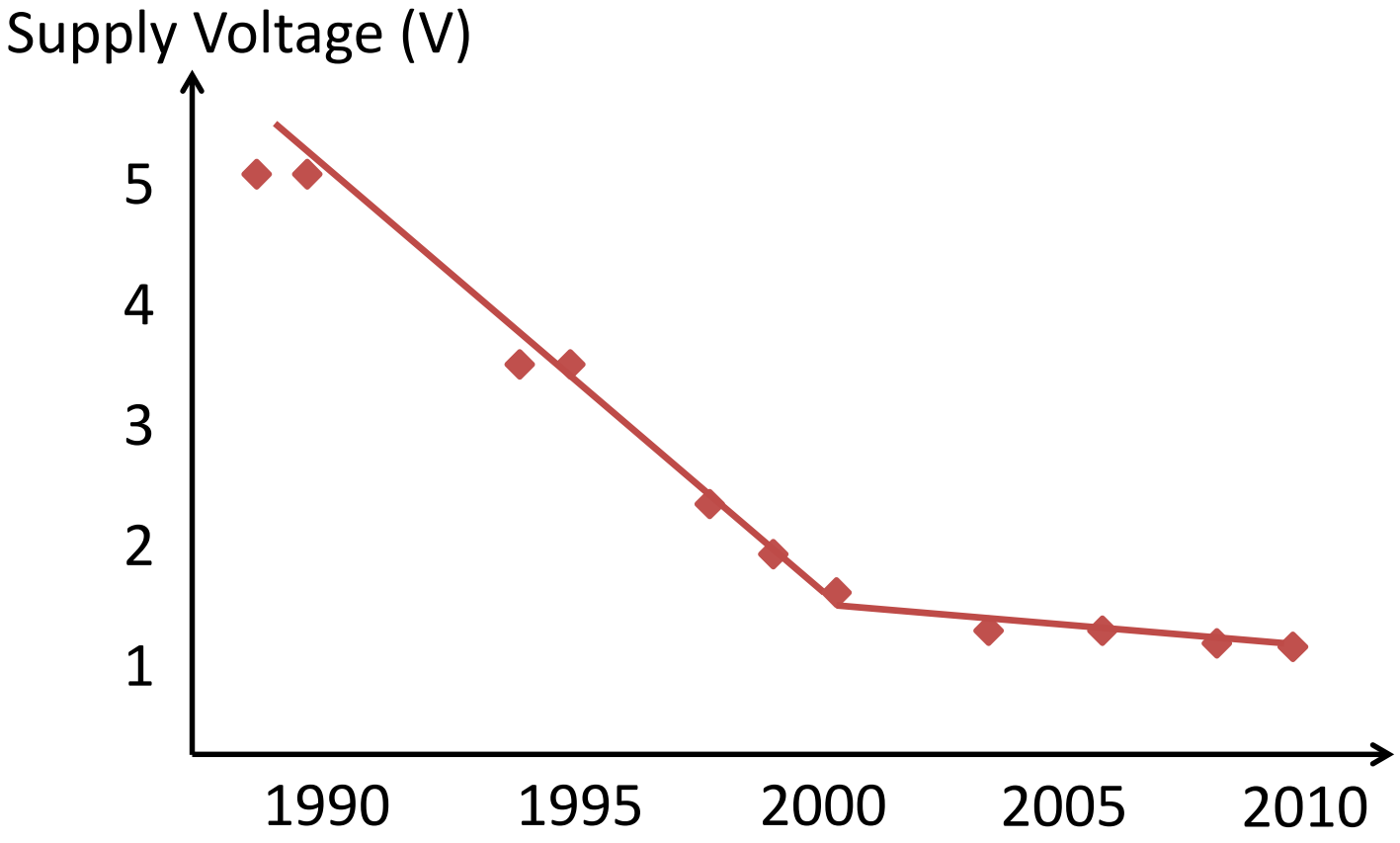


½ CV^2 stored in capacitor C is consumed on device resistor

Total energy consumed = ½ CV^2 + ½ CV^2 = CV^2

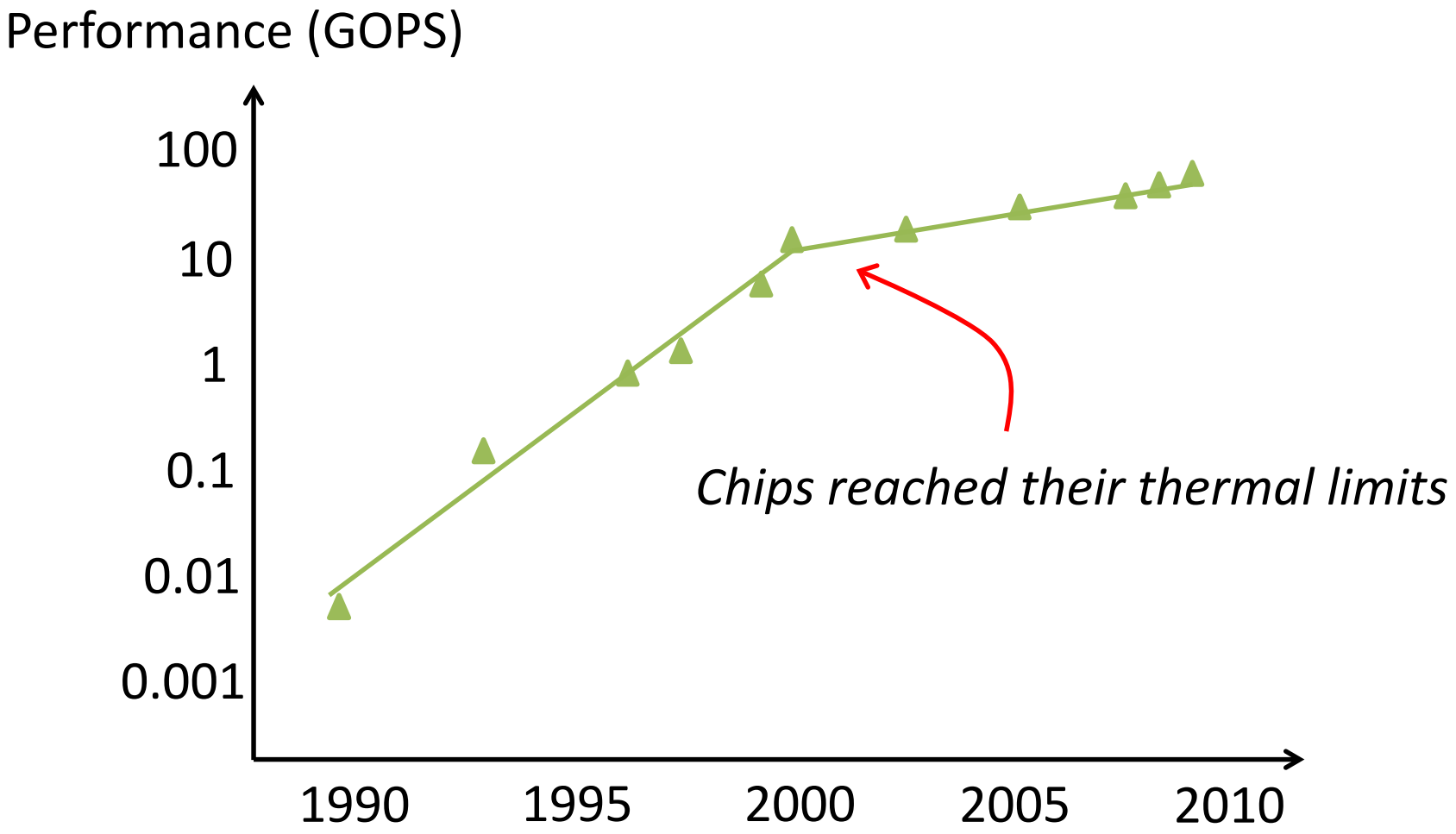
Voltage Scaling

Post-2000



Performance Scaling

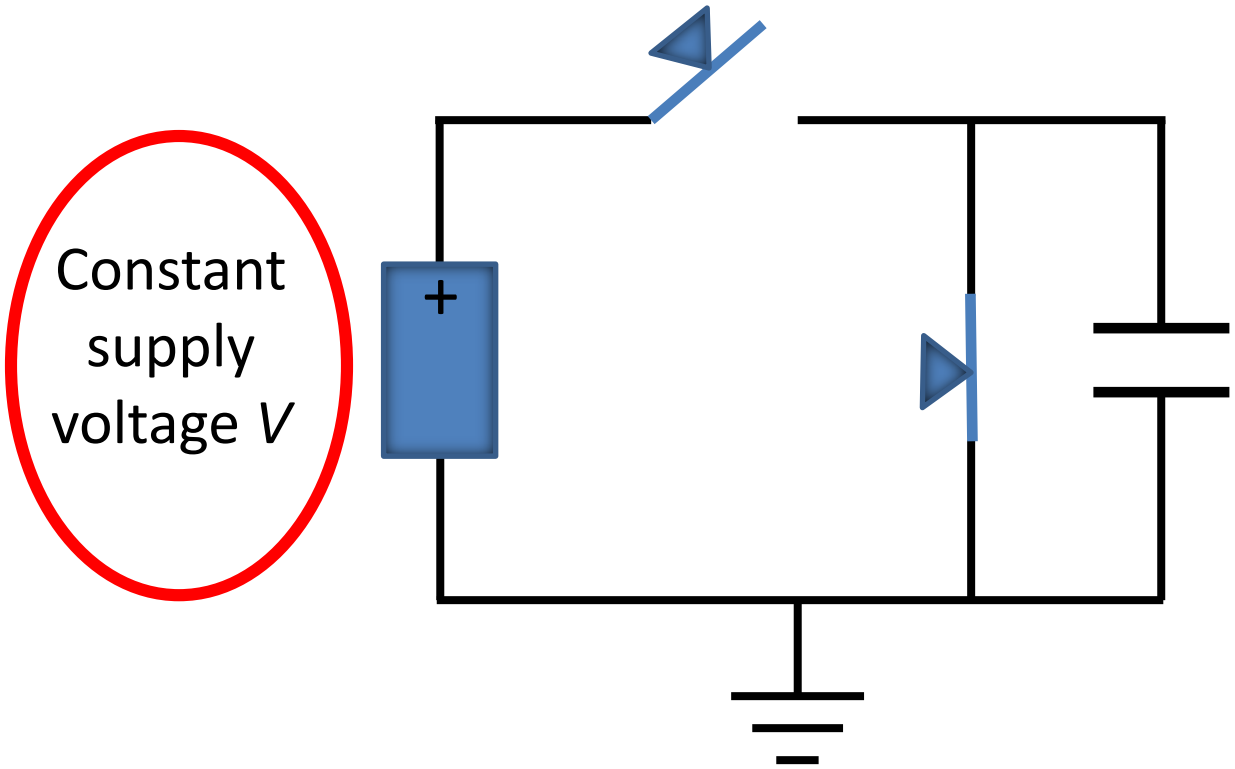
Post-2000



Now What?

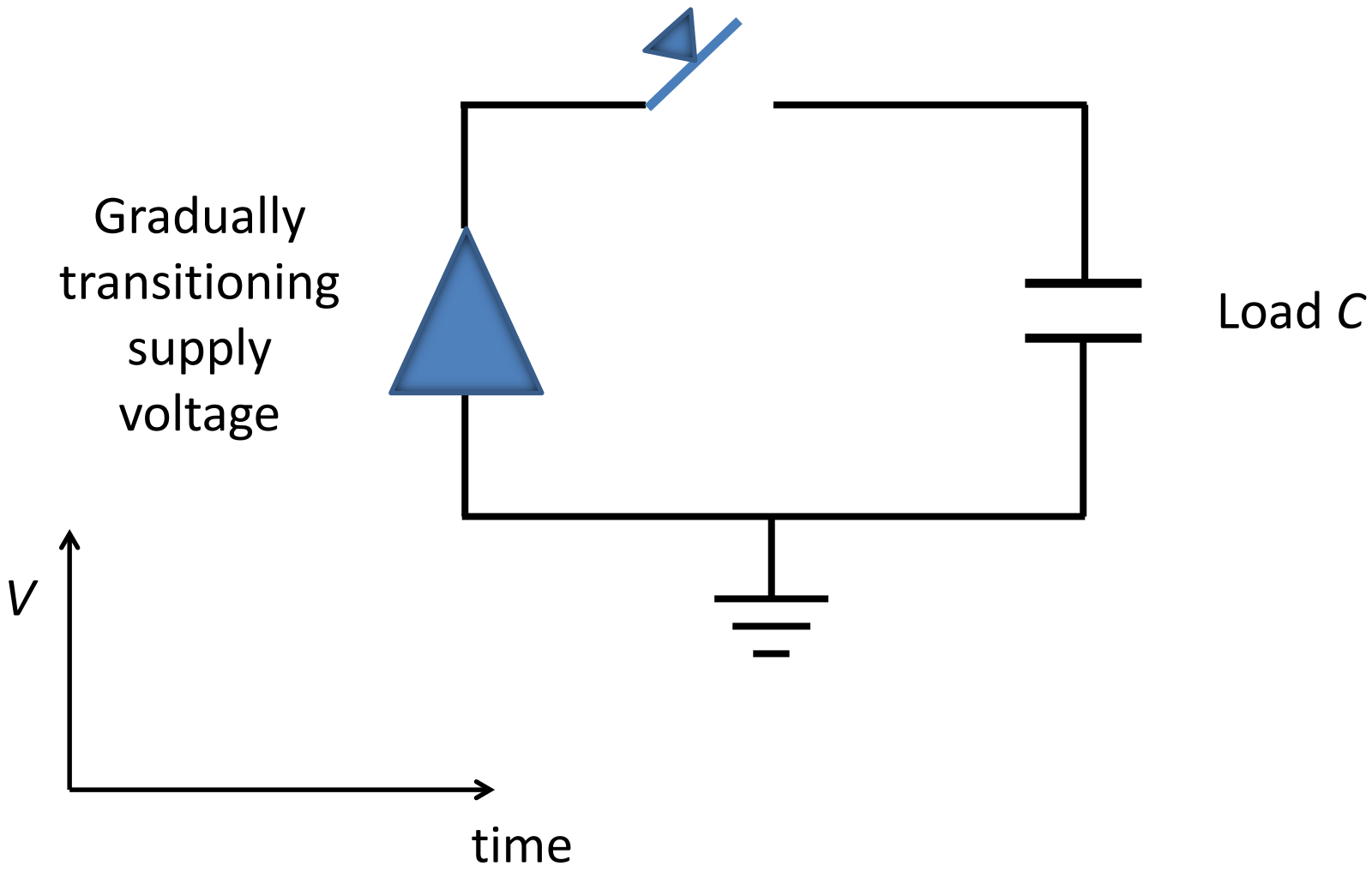
- 2000-present: Let's turn off whatever is not used
  - Clock gating
  - Power gating
  - Multi-core architectures
  - Software-controlled power management
- Limited ability to scale down server power with utilization: Deeper power down → longer idle-to-active → delayed response times
- At the end of the day, we are basically stuck!

Fundamental Physical Limitation



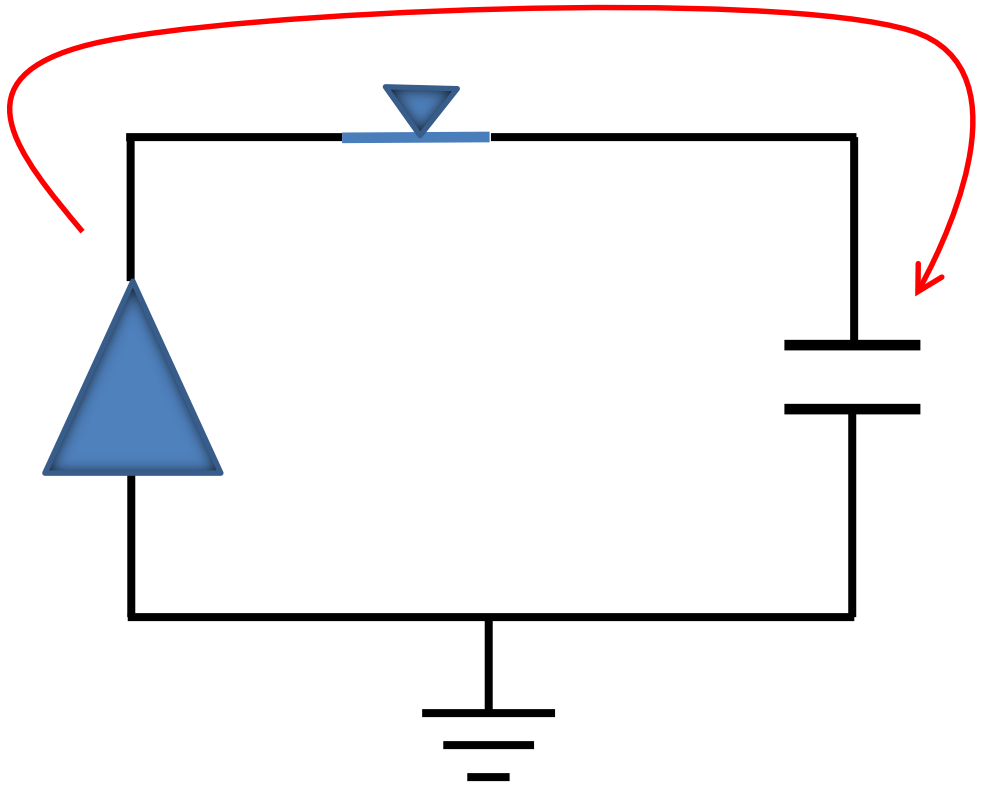
Total energy consumed = CV^2

Let's Modify the Voltage Supply



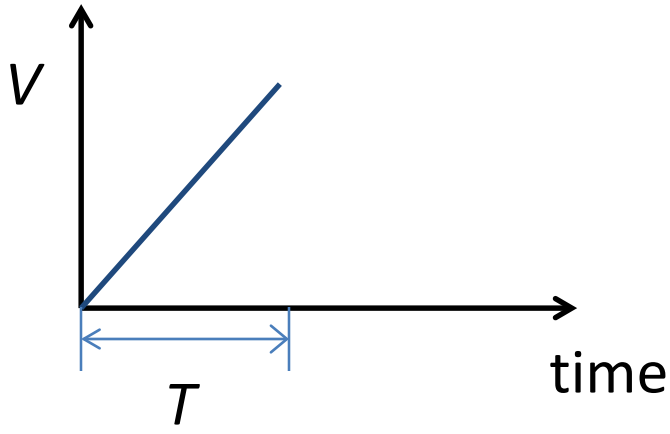
Energy Consumption for Gradual Charging

(RC/T) CV^2 consumed on R of device



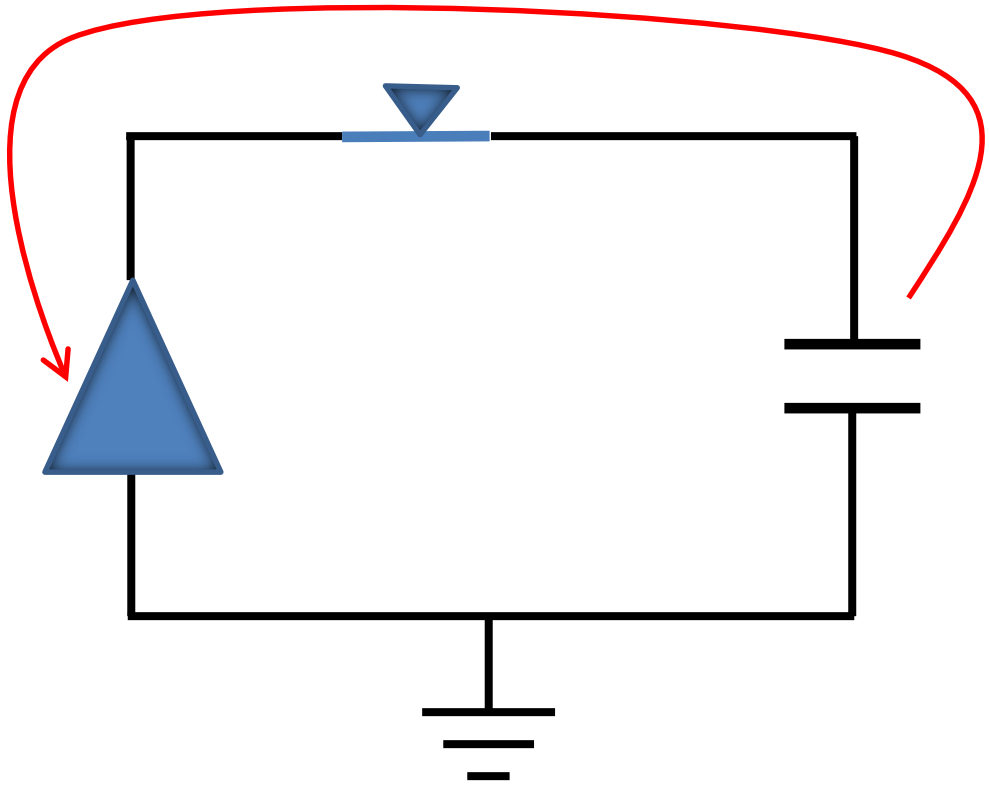
Gradually transitioning supply voltage

½ CV^2 stored in load C



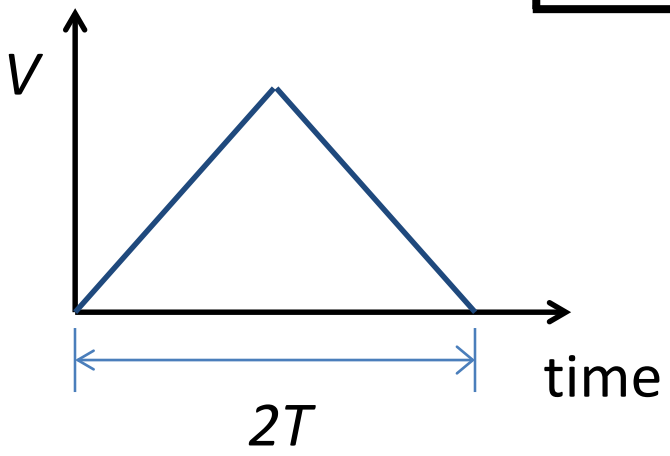
Energy Consumption for Gradual Discharging

(RC/T) CV^2 consumed on R of device



Gradually transitioning supply voltage

½ CV^2 stored in C returns to supply



Putting it All Together

Total energy consumed to charge/discharge

$$\sim= \quad 2 (RC / T) \quad CV^2$$

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 re-produce the initial state of the computing machine)

- Argument based on statistical thermodynamics:
  Loss of information → change in system entropy
  → energy converted to heat (i.e., wasted)

Reversible Turing Machines and Universal Gates

- Logically reversible Turing Machines (Bennett, 1973)
  - Compute just like a conventional TM, but keep intermediate results
  - Output final result
  - Reverse operation, disposing of intermediate results and returning the machine to its original state
- Reversible universal gates (Fredkin, Toffoli, 1982)
  - 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)
  - Focus on simple reversible function
- Gradual charge/discharge through inductors or capacitor “ladders”
- Issues
  - High overheads: Too many bits/gates
  - High complexity: Too many wires
  - 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.")

$$F(x) = x$$

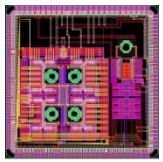
"Identity function"

a.k.a.

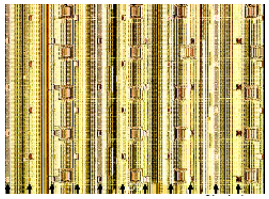
"clock" (in synchronous digital systems)

Chip Designs with Resonant Clock Mesh Distribution Networks

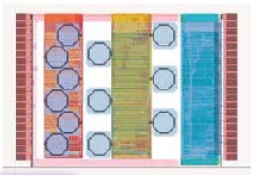
GHz clock
CICC'06



IBM Cell BE
ISSCC '09



2GHz FPU
A-SSCC '10



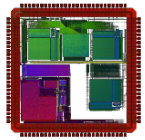
IBM Power8
ISSCC '14

Cyclos Semiconductor

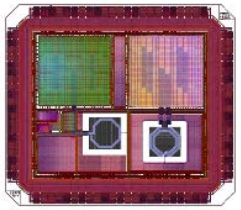
AMD Piledriver
ISSCC '12

IBM System z
ISSCC '15

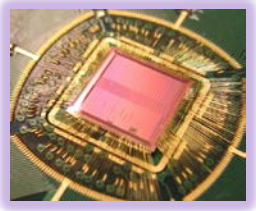
AMD Steamroller
ISSCC '15



DWT ASIC
ISLPED'03

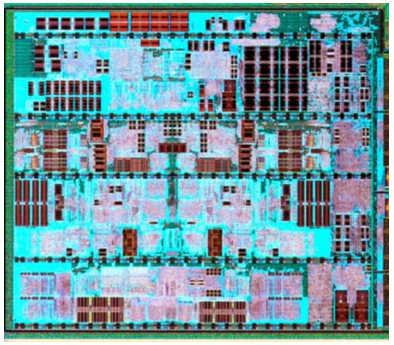


GHz FIR
VLSI'07
CICC'07



ARM 926EJ-S
ESSCIRC '09

Cyclos Semiconductor



2003

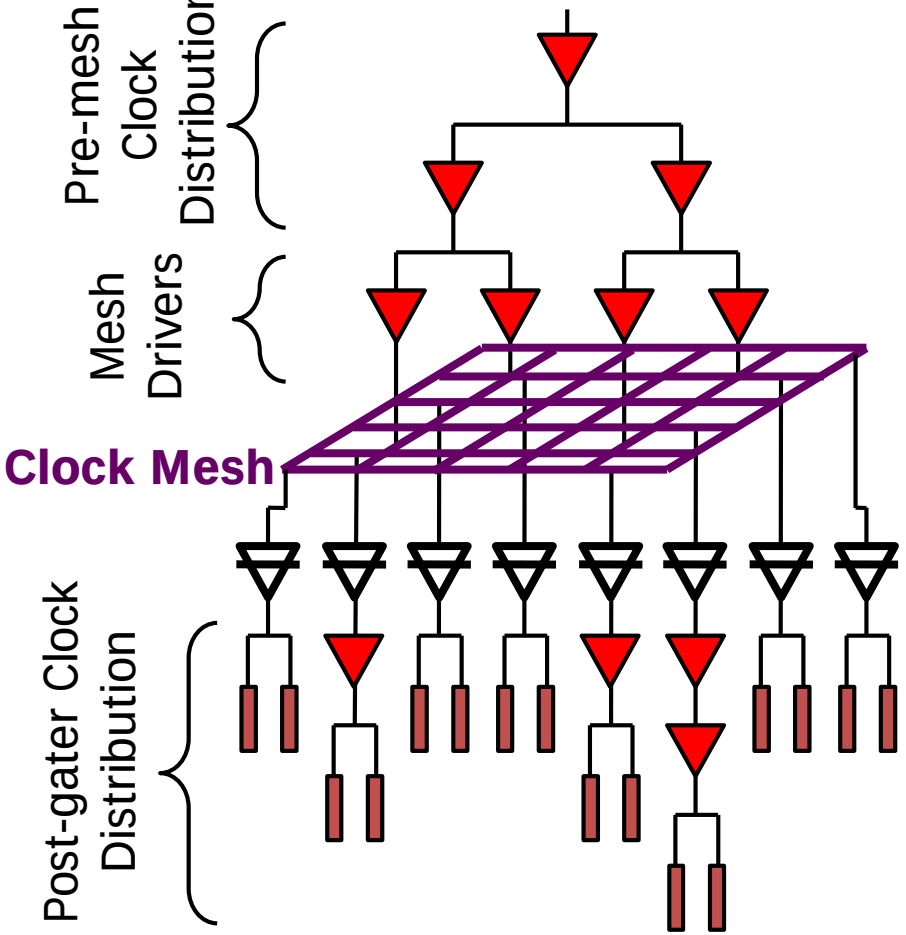
2015

200MHz
250nm
area < 1mm^2
silicon prototypes

5+ GHz
22nm
area > 20mm^2
high-volume commercial CPUs

Clock Mesh

Typical solution for multi-GHz processors



Pros

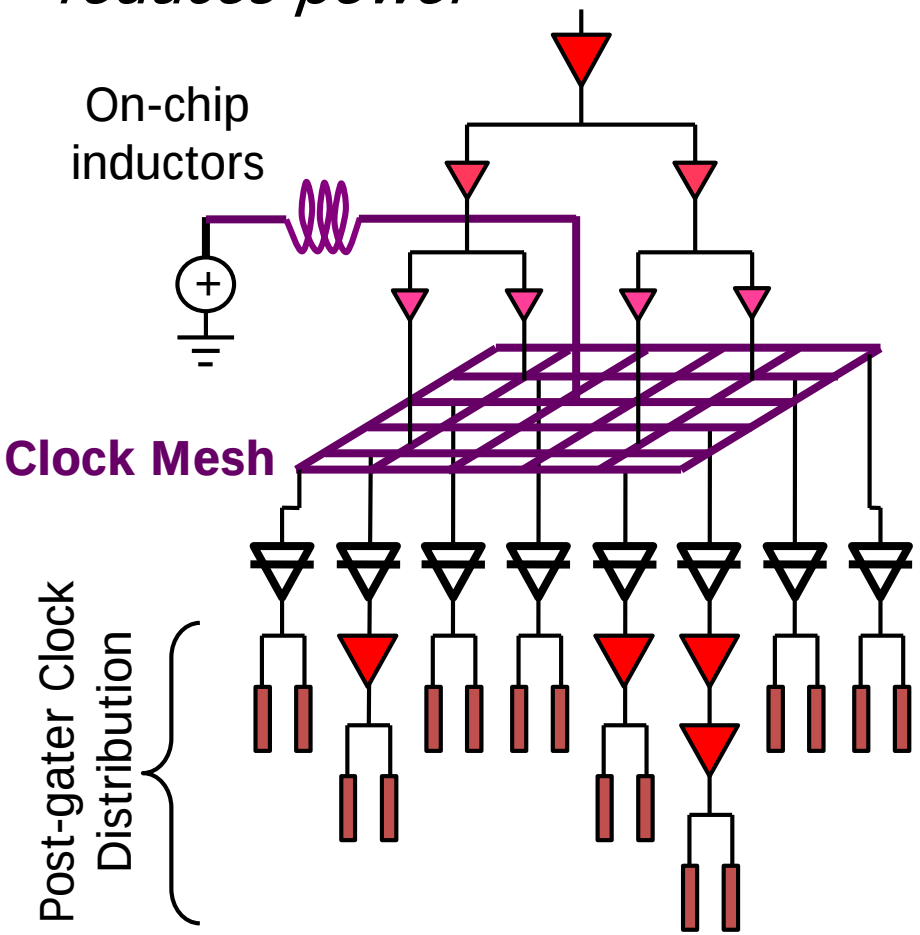
- All-metal mesh shorts clock buffer outputs → very low skews
- Mesh isolates local timing → simplifies late-design ECOs

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

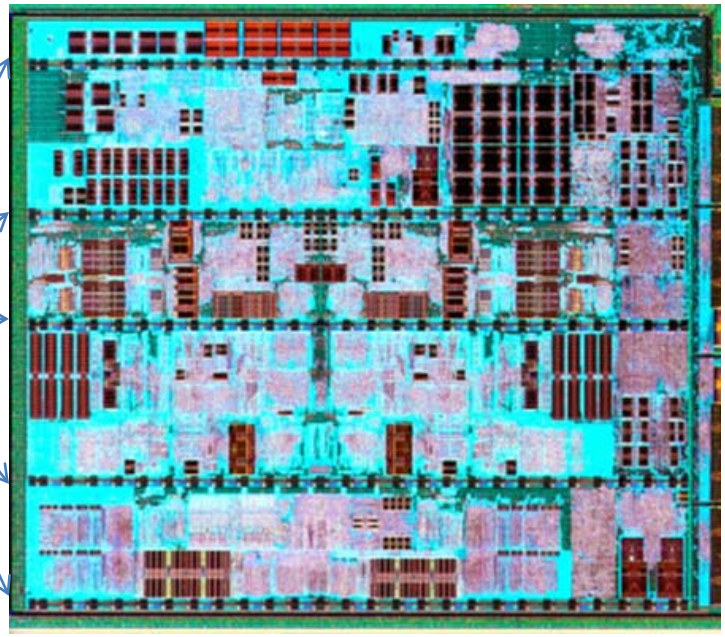
- All-metal mesh shorts clock buffer outputs → very low skews
- Mesh isolates local timing → simplifies late-design ECOs

Cons

- ~~Large capacitance of clock mesh leads to high power consumption~~

4+ GHz x86 “Piledriver” Core

5 rows with ~20 inductors each

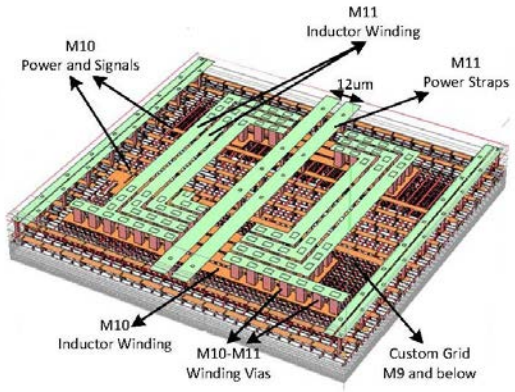


AMD

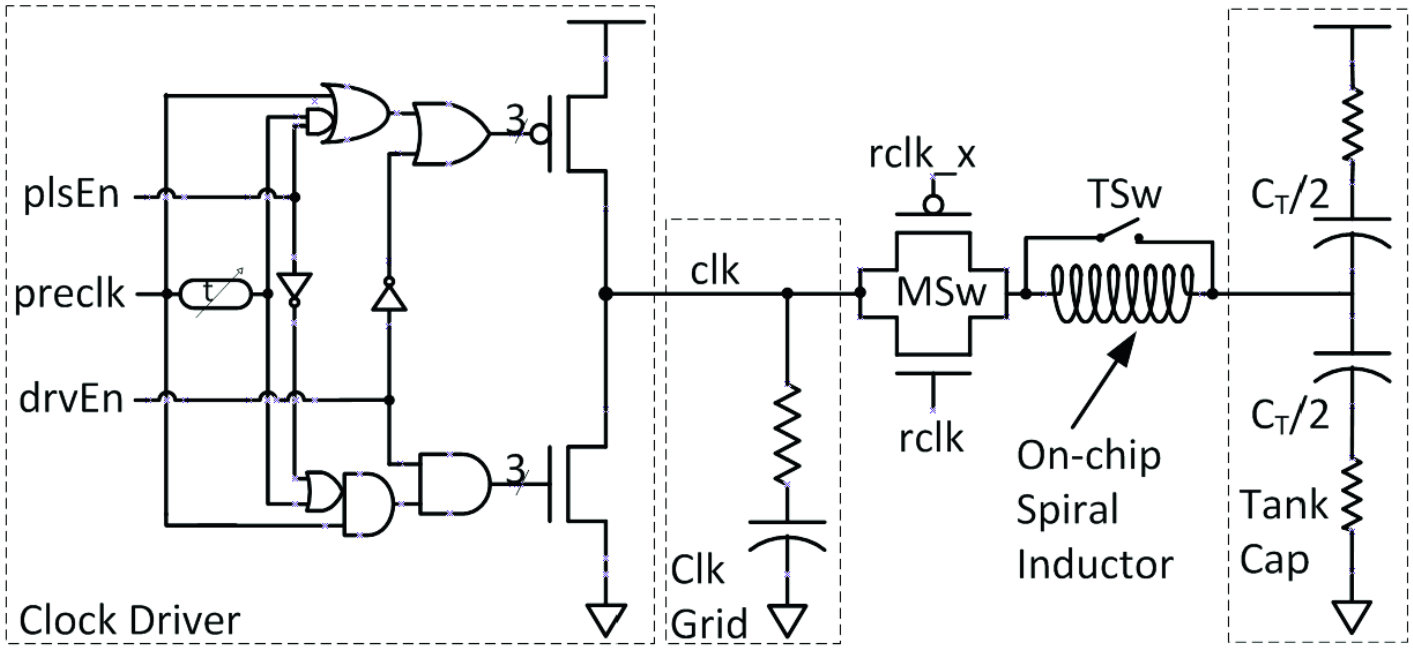
Cyclos Semiconductor

Sathe *et al.*, ISSCC ‘12

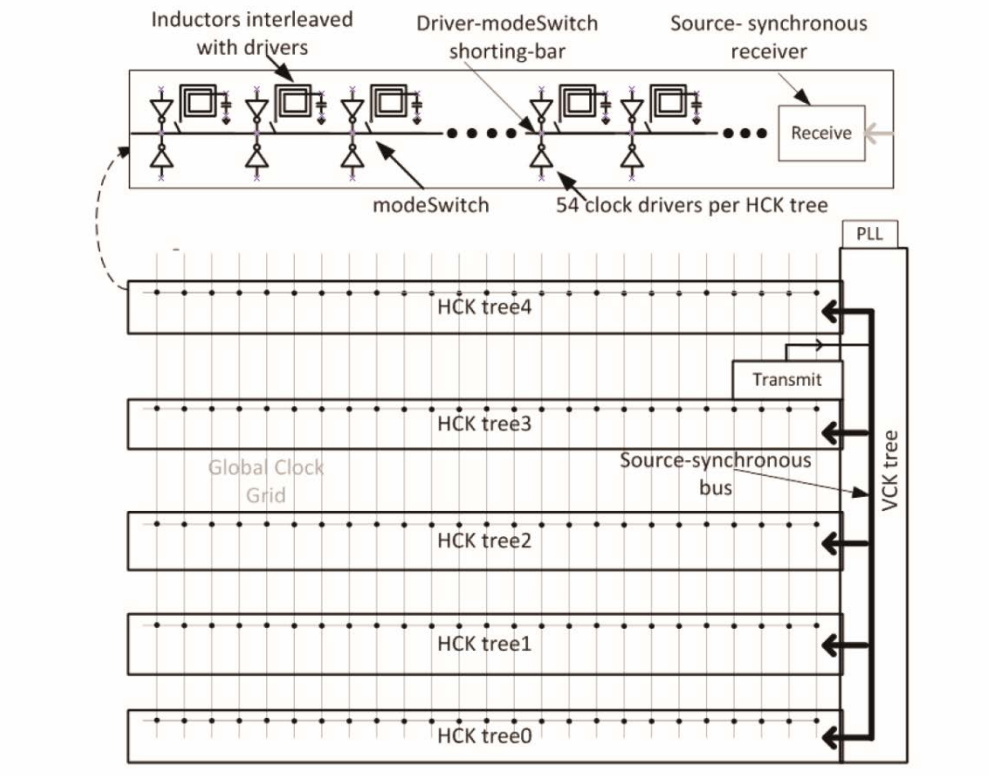
- High-volume 64-bit x86 core in 32nm, 11 metal layers
- 100 on-chip inductors, 0.7nH to 1.2nH
- Up to 30% clock power savings
- 5% to 10% reduction in total chip power, depending on workload profile



Simplified Model of Dual-Mode Clock Network

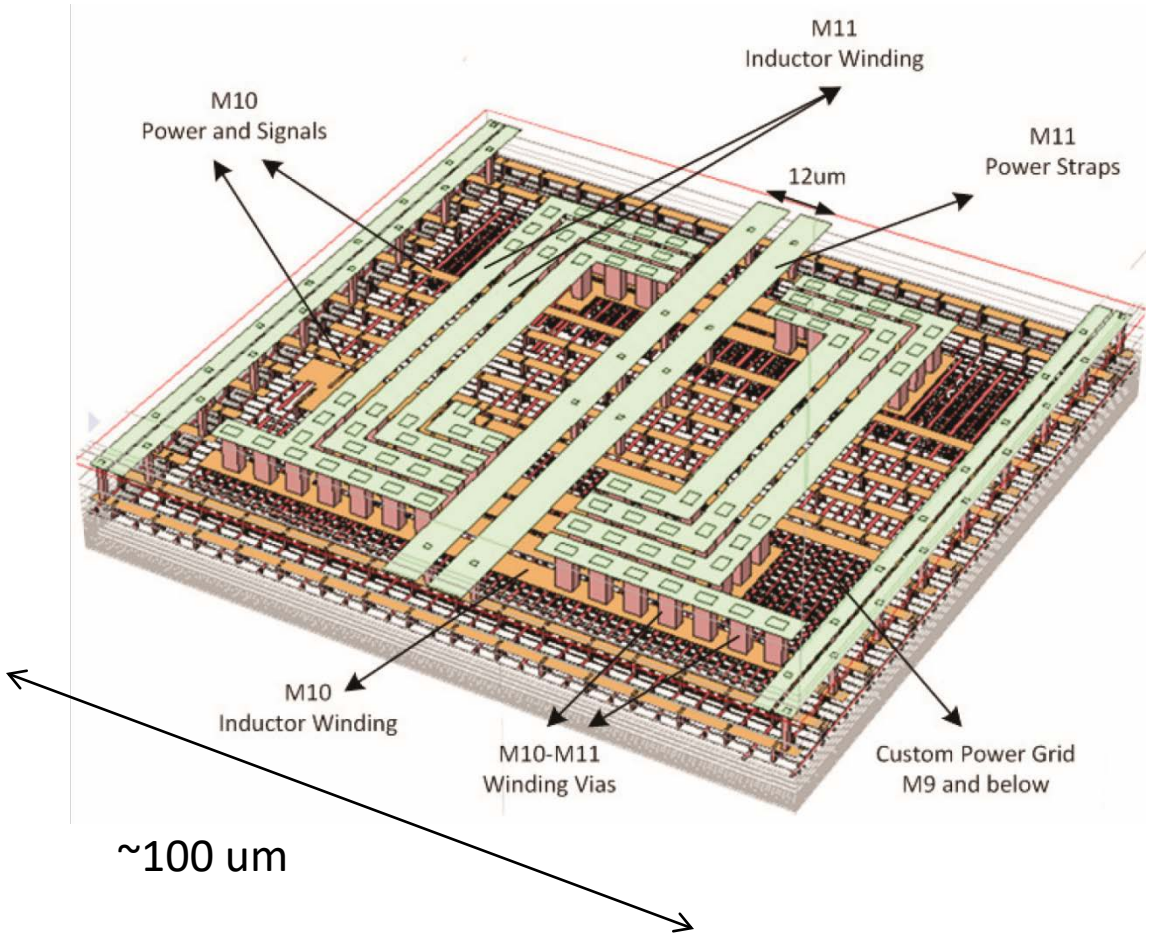


Global Clock Organization and Distribution

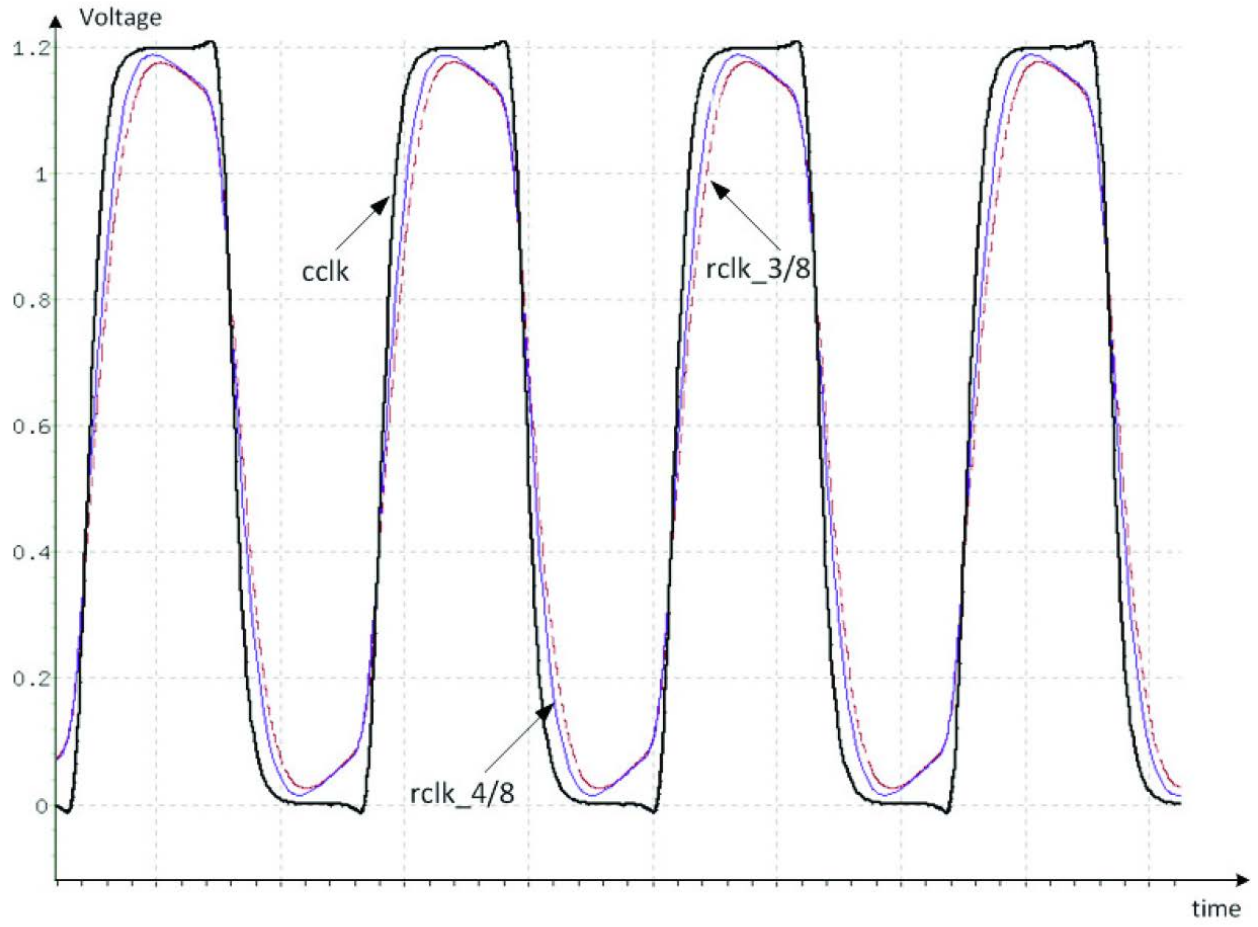


- Large variation of capacitance across the chip
- Palette of 5 inductors (0.7nH to 1.2nH)
- Integer linear program to select inductor values and size clock wires to minimize clock skew

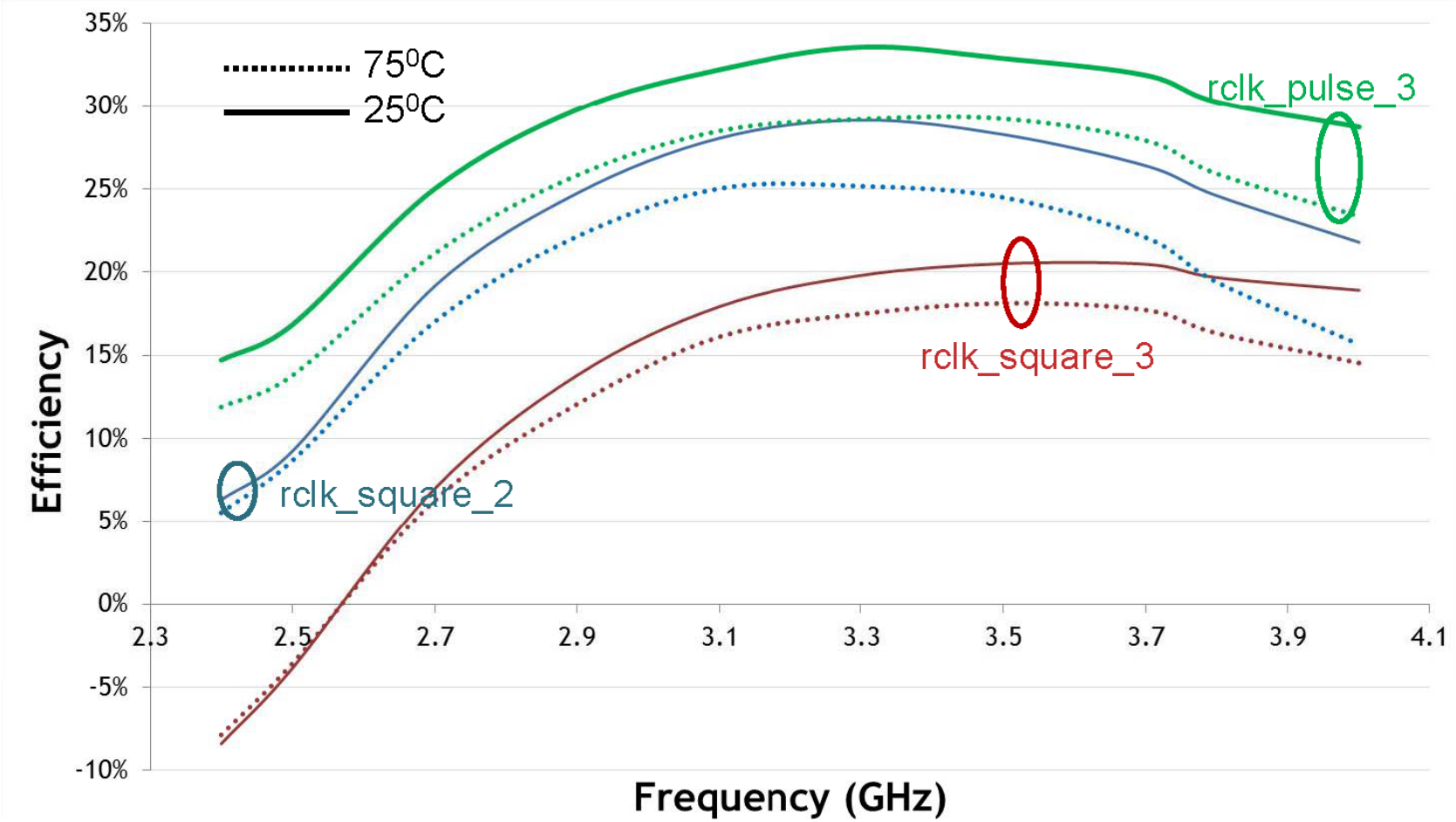
Inductor Design



Simulated Clock Waveforms

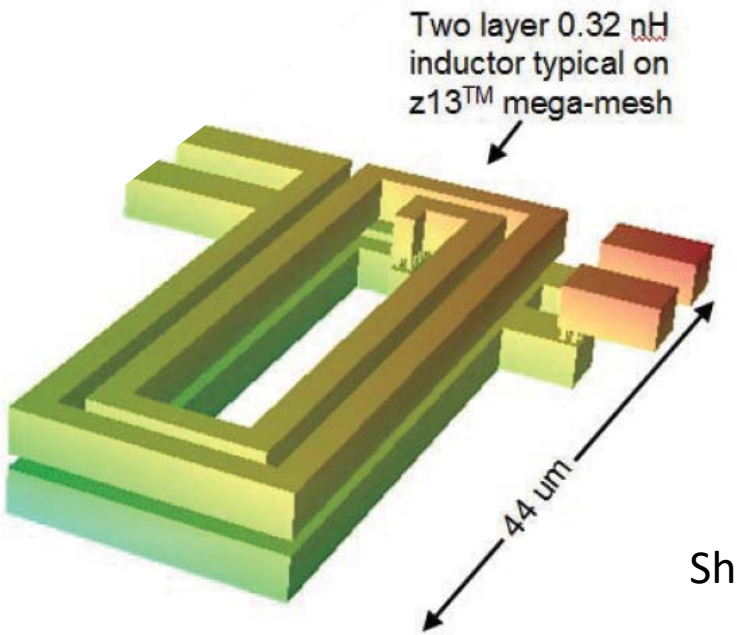


Clock Power Savings vs Frequency



- Efficiency : Percentage clock power savings over cclk
- rclk_square_x → Clock driver strength modulation of x/7

Resonant Mega-Mesh for IBM z13™



Shan *et al.*, VLSI Symp 2015

- 545mm^2 (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

How About Logic?

Total energy consumed to charge/discharge

$\sim= \quad 2 (RC / T) \quad CV^2$

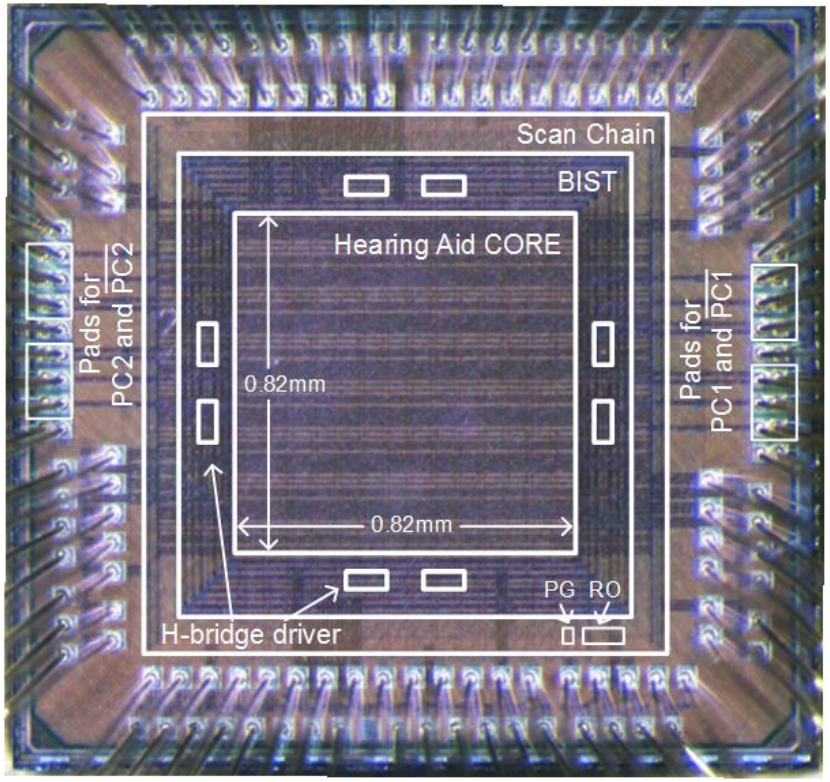
Gradual recycling of charge saves energy (assuming RC/T < 2)

True for all-metal clock networks, but is it true for logic?

In fact, the slower the better

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

13.8µW Binaural Dual-Microphone ANSI S1.11 Filter Bank for Hearing Aids



Wu *et al*., ISSCC 2017

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

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