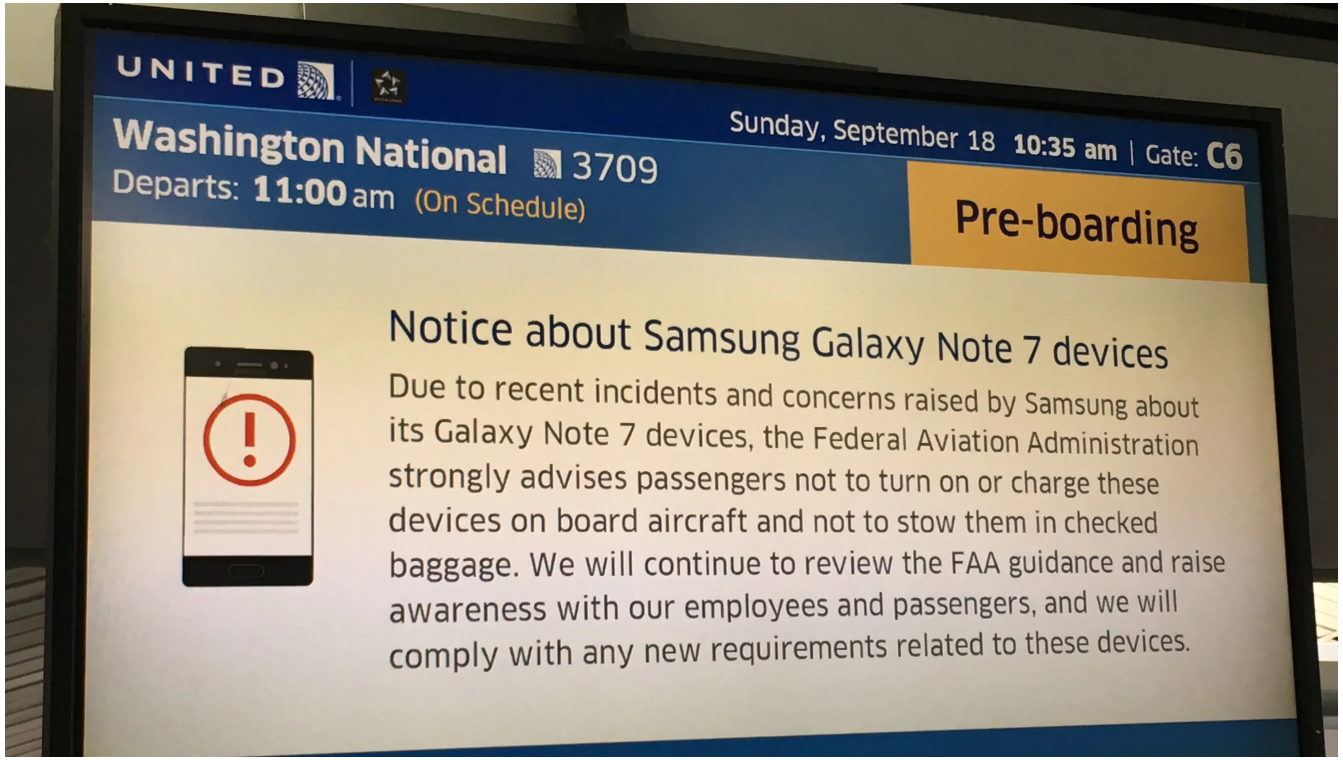
Why save power?

- Power/Energy is important, especially for embedded systems
  - Mobile and portable devices: Smartphones, Laptops
  - Other devices: Wearables, Medical devices, etc.
  - 3D architectures
- Lower power usage = Better battery life
- More power = More heat

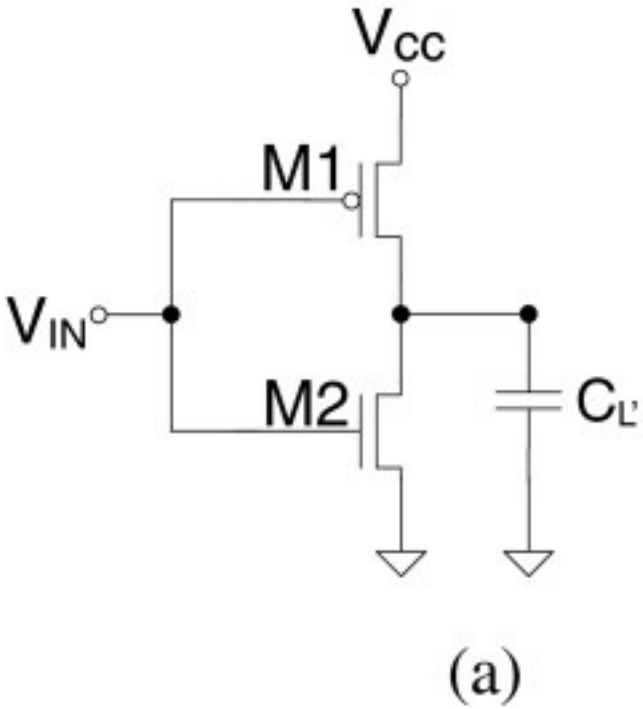
Samsung Galaxy Note 7



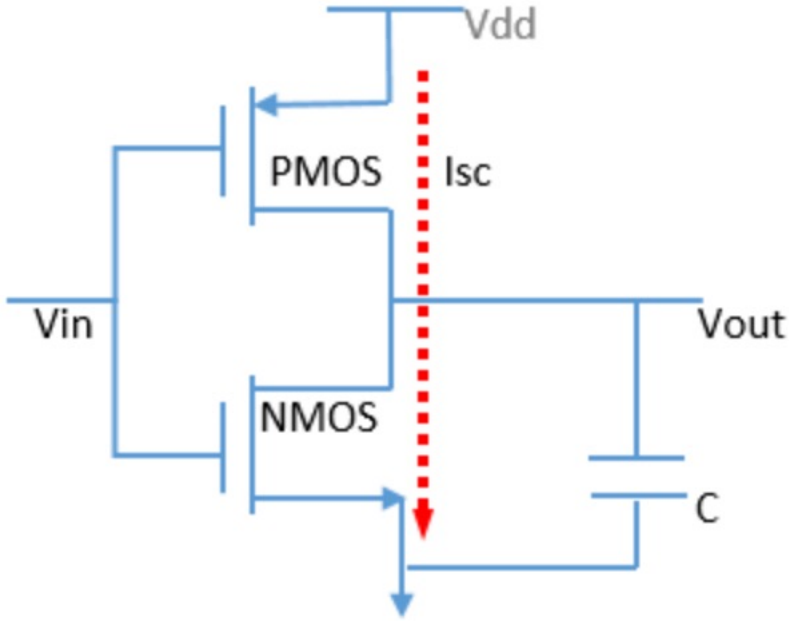
https://www.theverge.com/2016/10/9/13218730/samsung-galaxy-note-7-fire-replacement-fourth-virginia
https://www.theverge.com/2016/10/9/13215728/samsung-galaxy-note-7-third-fire-smoke-inhalation
https://mashable.com/article/samsung-galaxy-a21-airplane-fire

Some power basics

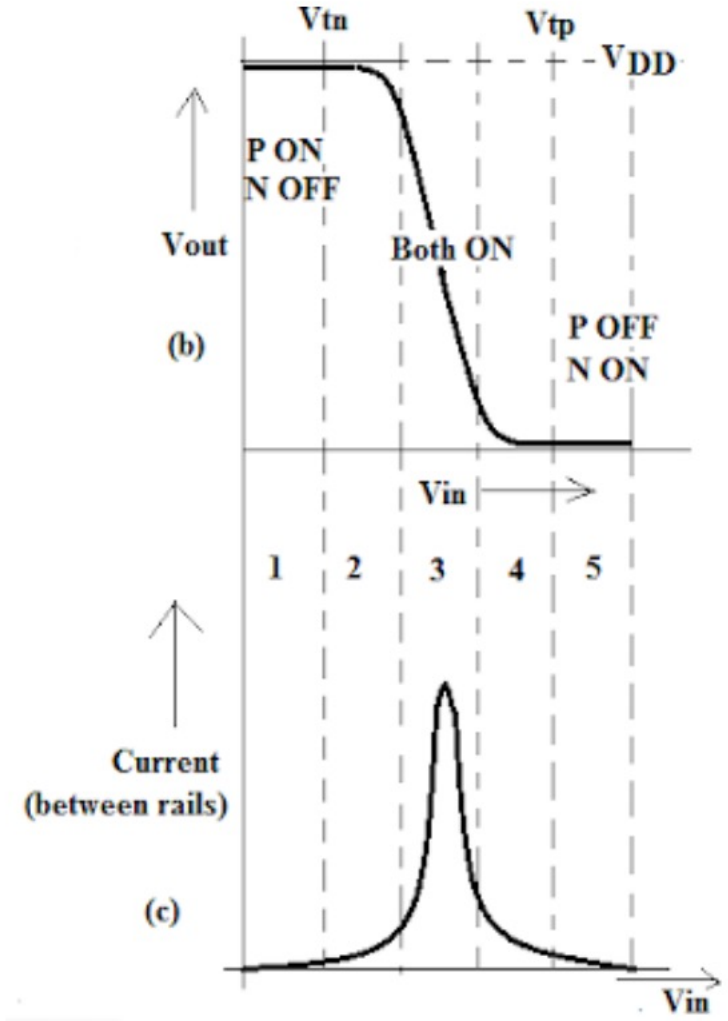
- Processors have gates
  - Short circuit power + Leakage power + Dynamic power
  - $P_{dynamic} = \alpha CV^2f$ (P: power, α: switching factor, C: capacitance, V: voltage, f: frequency)
- E = P * Execution time



Short circuit power



SHORT CIRCUIT CURRENT PATH IN CMOS

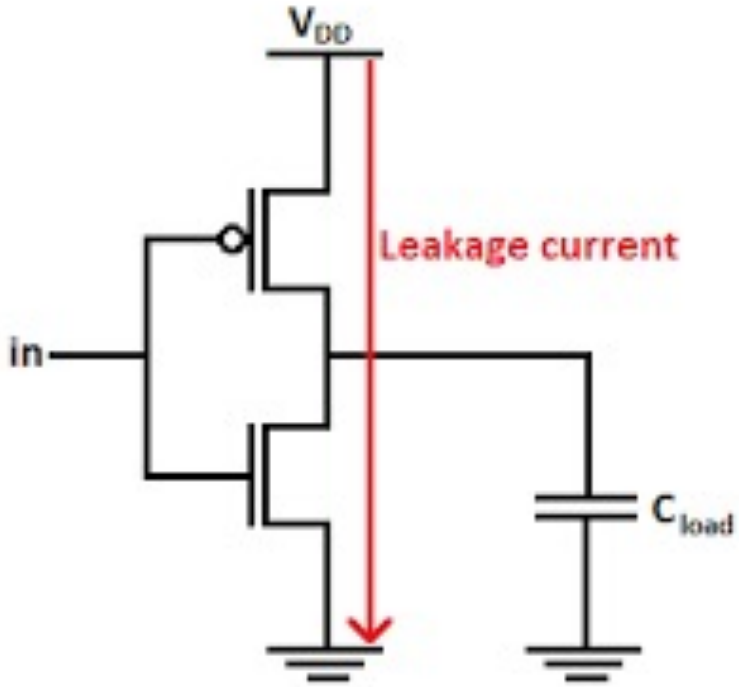


- Happens during transitions
- Both transistors momentarily ON

when both transistor are ON momentarily the short circuit comes in the form of spike

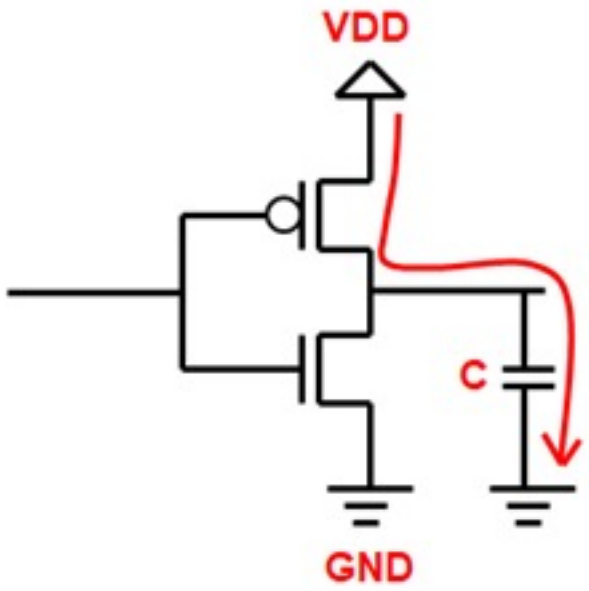
Leakage power

- Happens during normal operations
- OFF transistors are not ideal open circuit
- More serious for smaller process nodes

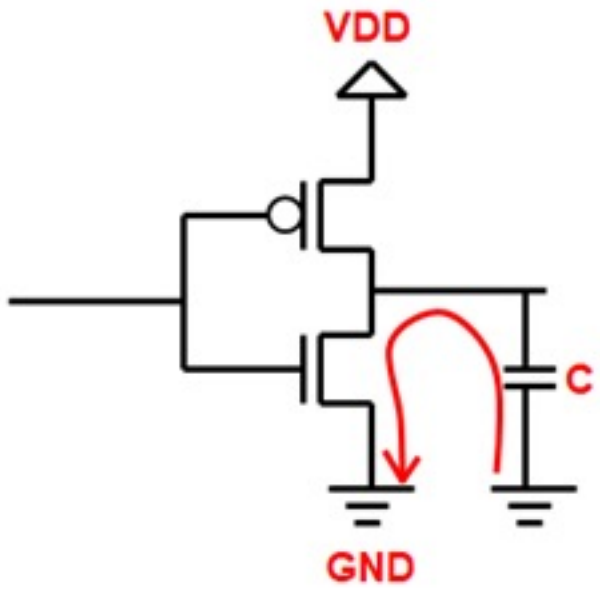


Dynamic power

- Happens due to switching
- Requires two toggles
  - Charge Load capacitor
  - Discharge load capacitor



During charging



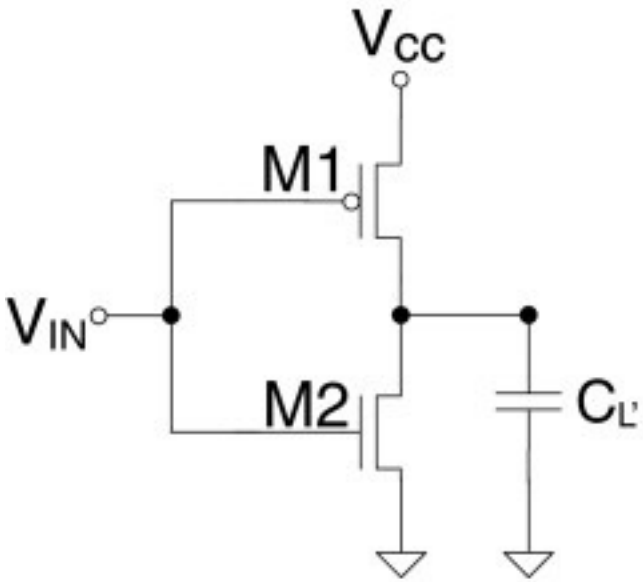
During discharging

How to reduce dynamic power?

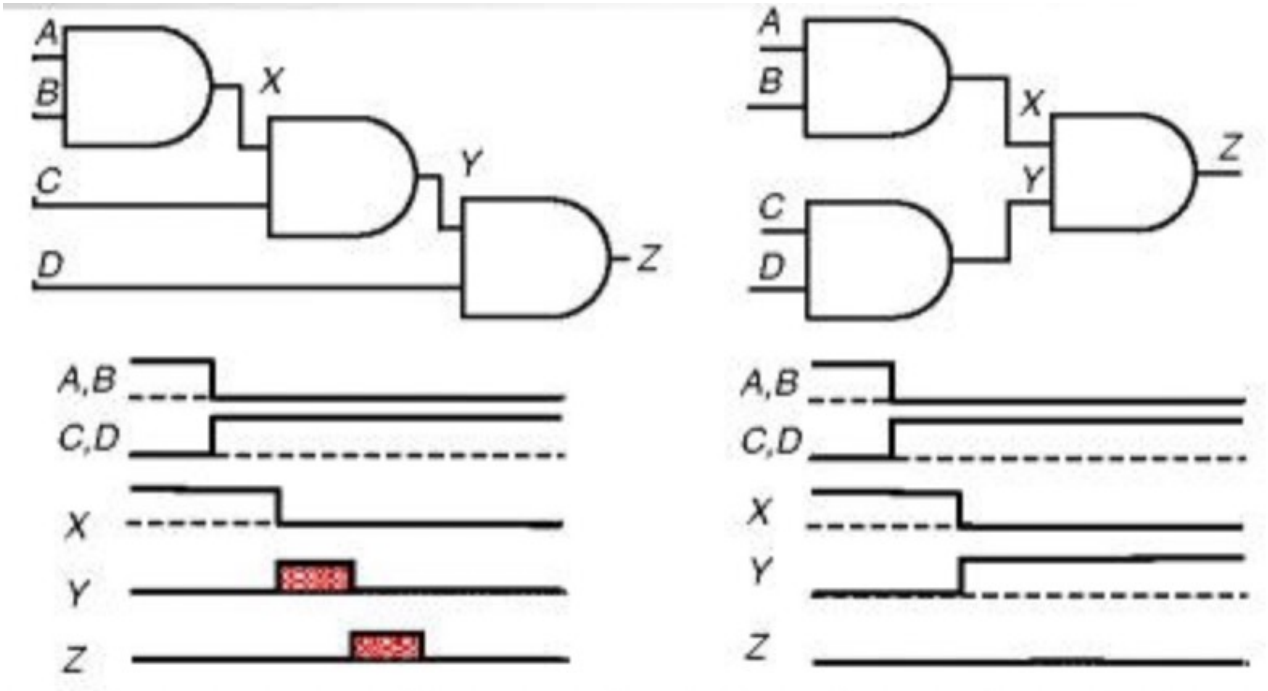
$P = \alpha C V^2 f$

$\alpha \downarrow$, $C \downarrow$, $V \downarrow$, $f \downarrow$ => $P \downarrow$

- Reducing α: design circuit with suitable gates, reduce glitch, power gating
- Reducing C: use smaller gates
- Reducing V: use lower supply voltage
- Reducing f: use lower clock frequency
- Reducing V has the highest benefit

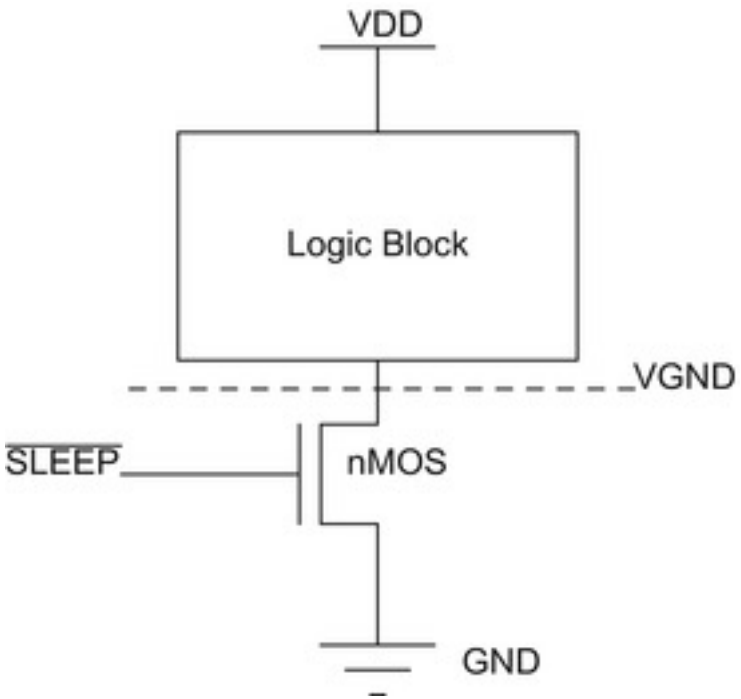


Avoid glitches

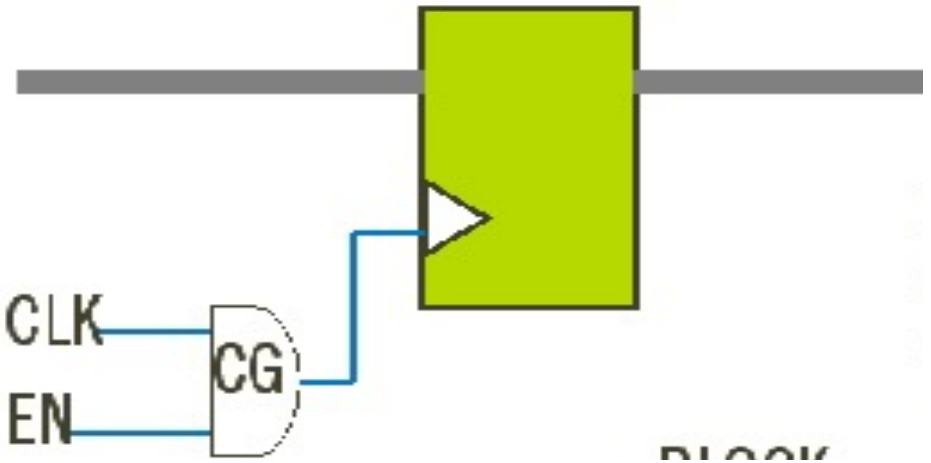


- **Unbalanced paths causes unnecessary toggles due to glitch**
- **Extra dynamic power consumed**
- **Creating balanced paths reduce α**
- **Low power dissipation**

Power and Clock gating



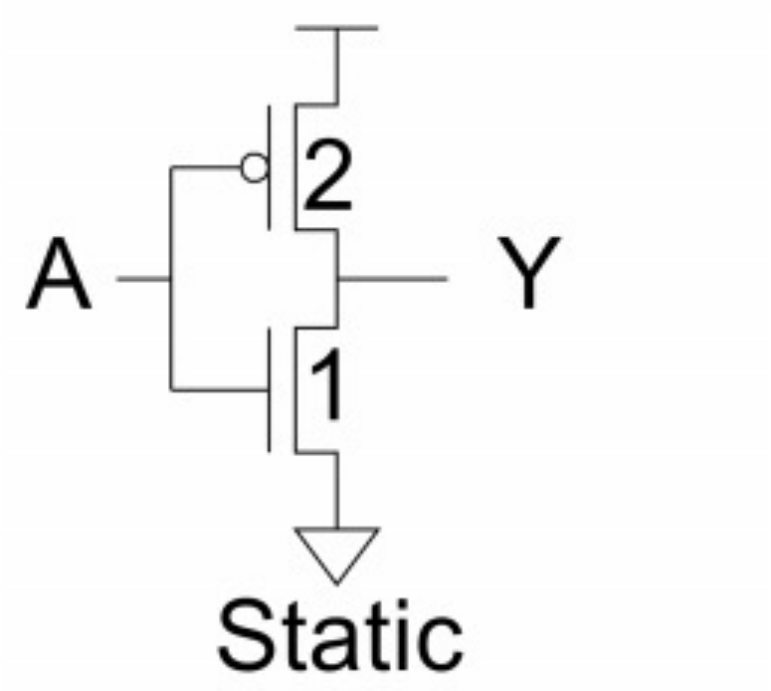
Power gating

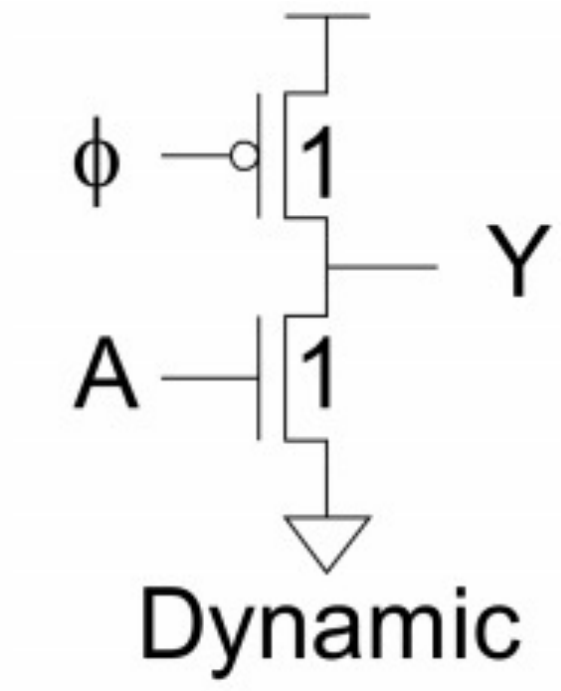


Clock gating

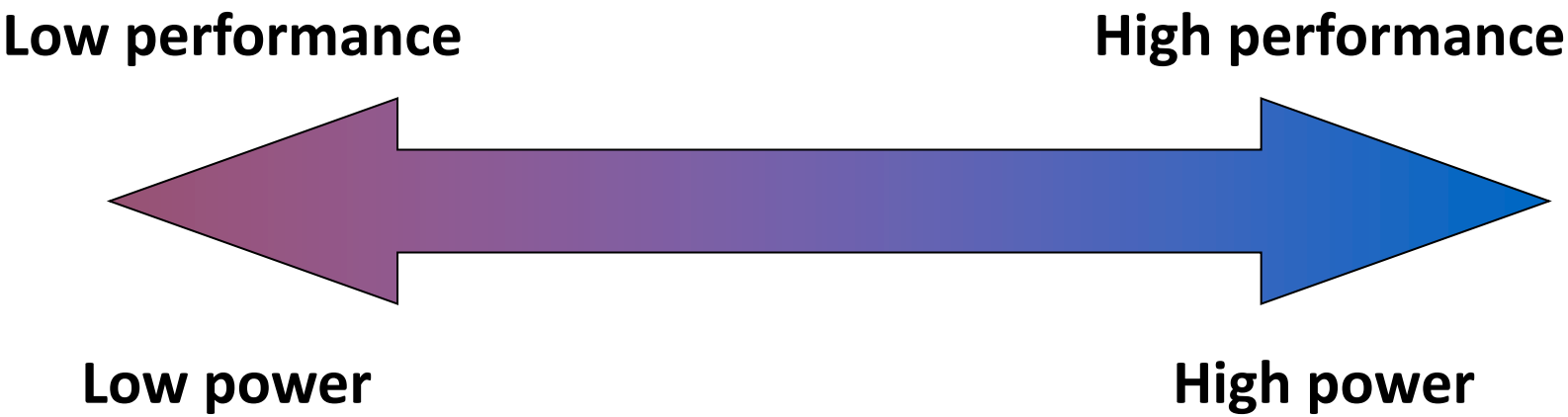
- **Power gating saves leakage power**
- **Clock gating reduces dynamic power**

Static vs Dynamic logic design





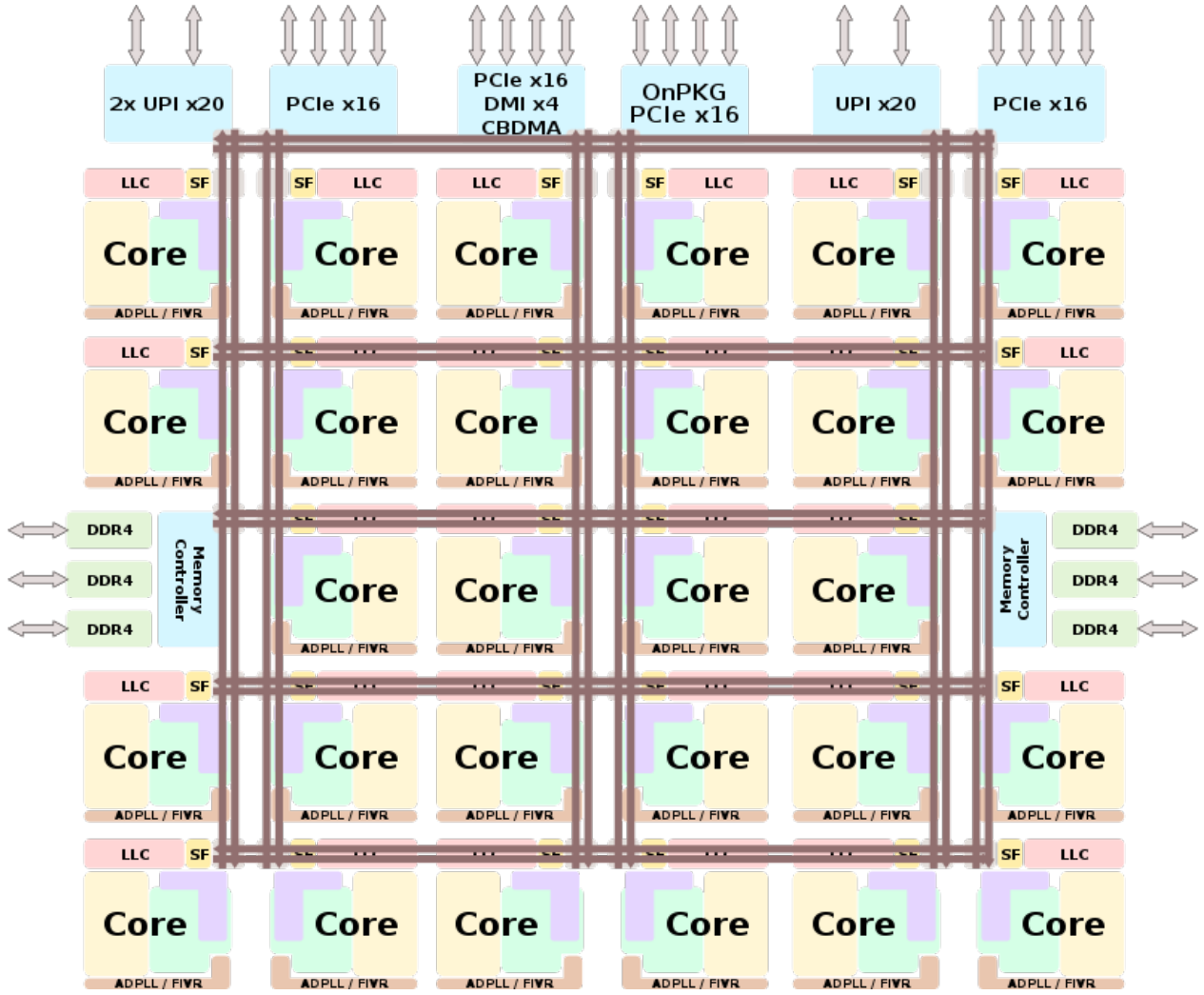
Power-Performance trade-offs



- Lower power:
  - Pros: Lower power bill, lower heat, lower fan noise, longer-lasting battery
  - Cons: Poor performance
- How about application-aware power management?

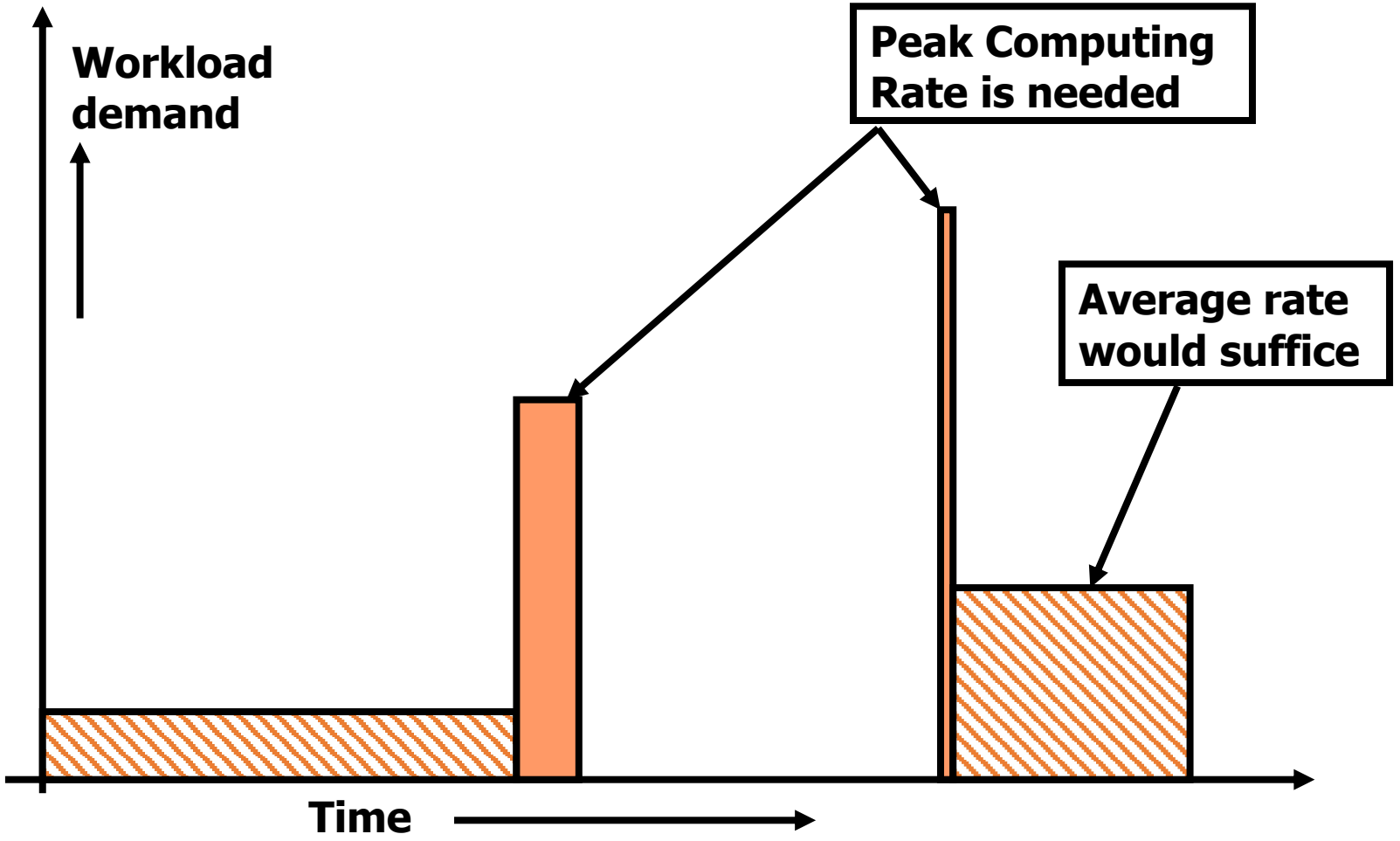
Saving power in Manycore

- Lots of cores
- Not all of them are used all the time
- Different levels of usage in each core
- Opportunity to save power

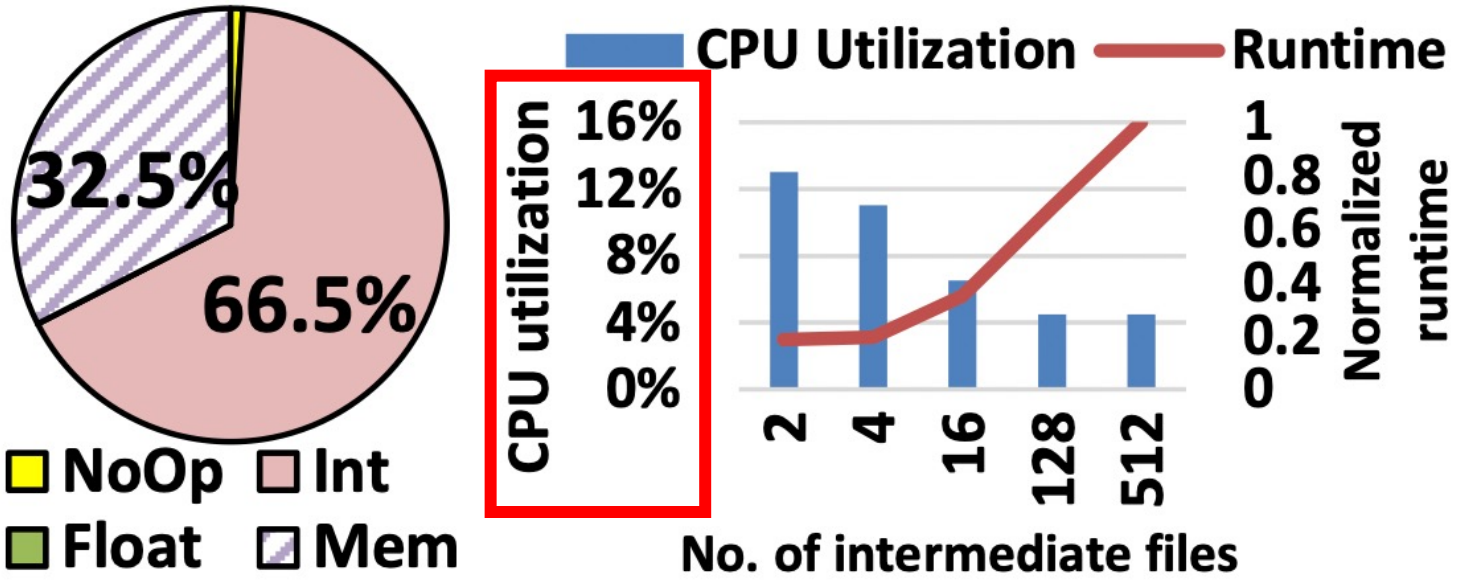


Workload characteristics

- Applications do not require peak computing capability all the time
- Reduce V, f when the demand is low
- Meet deadlines exactly instead of being too fast/slow



Workload characteristics: k-mer counting

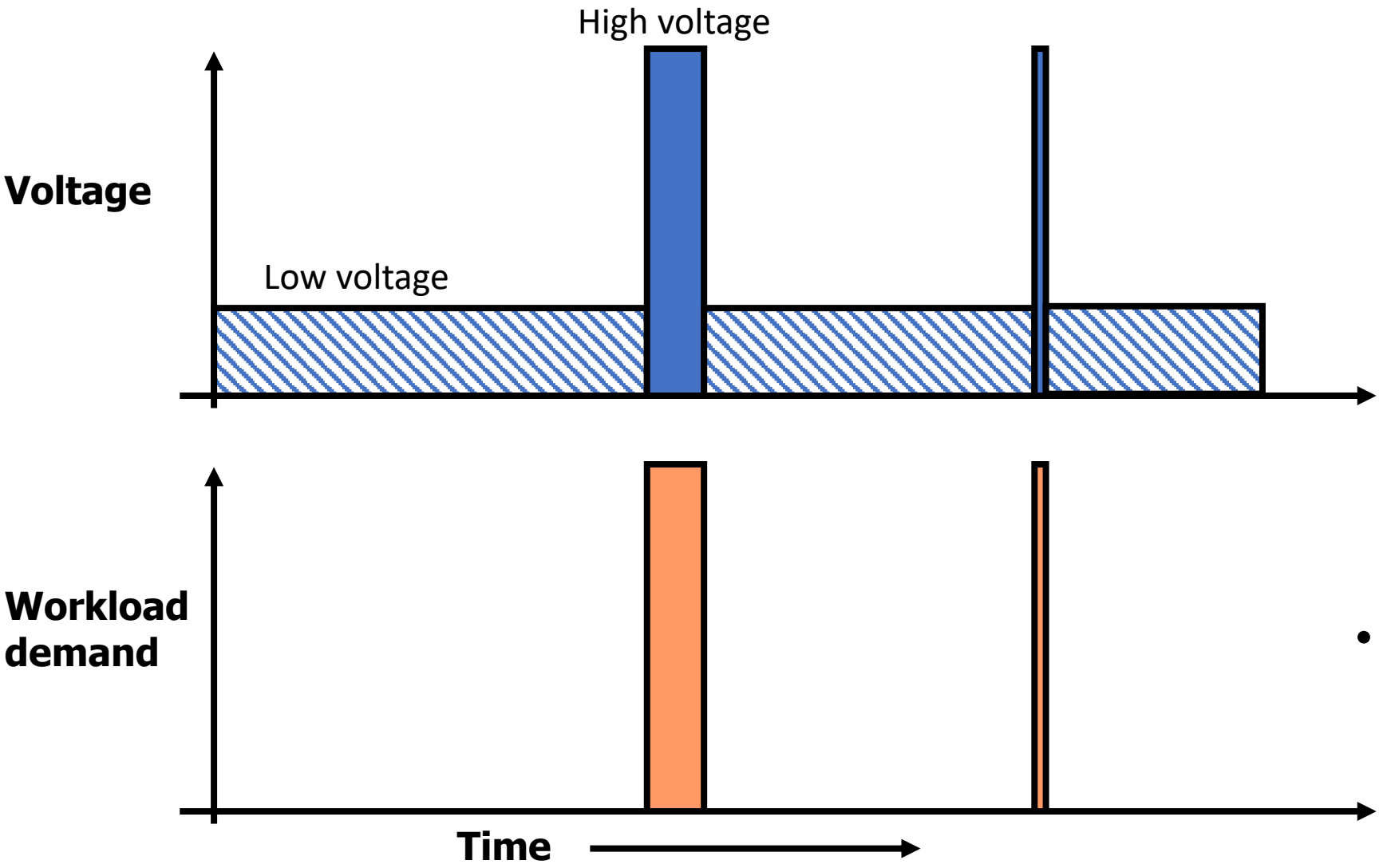


(a) (b)

Fig. 2: Gerbil: (a) Instruction types, and (b) CPU utilization and runtime (normalized) for varying number of intermediate files

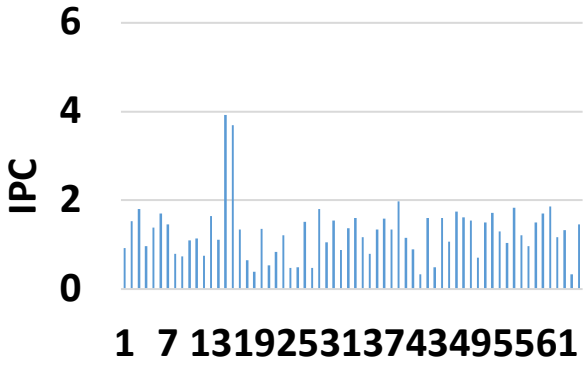
Biresh Kumar Joardar, Priyanka Ghosh, Partha Pratim Pande, Ananth Kalyanaraman, and Sriram Krishnamoorthy. 2019. NoC-enabled software/hardware co-design framework for accelerating k-mer counting. In Proceedings of the 13th IEEE/ACM International Symposium on Networks-on-Chip New York, NY, USA, Article 4, 1–8.

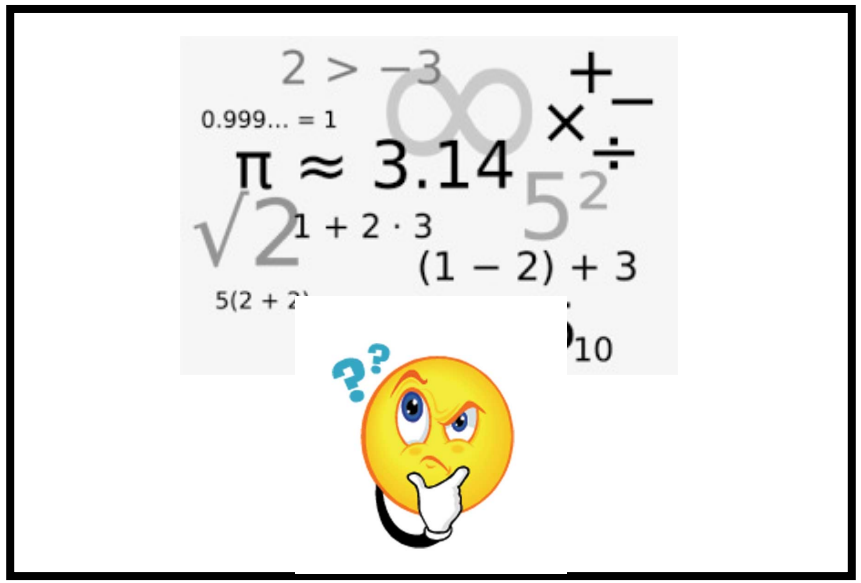
Voltage Frequency Scaling (VFS)



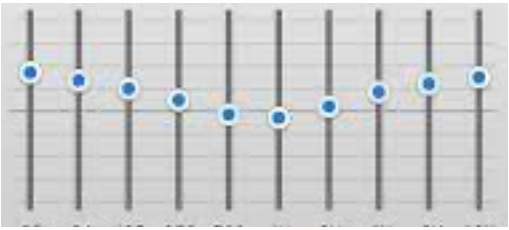
- **If you're not using something, turn it off**

Voltage Frequency Scaling



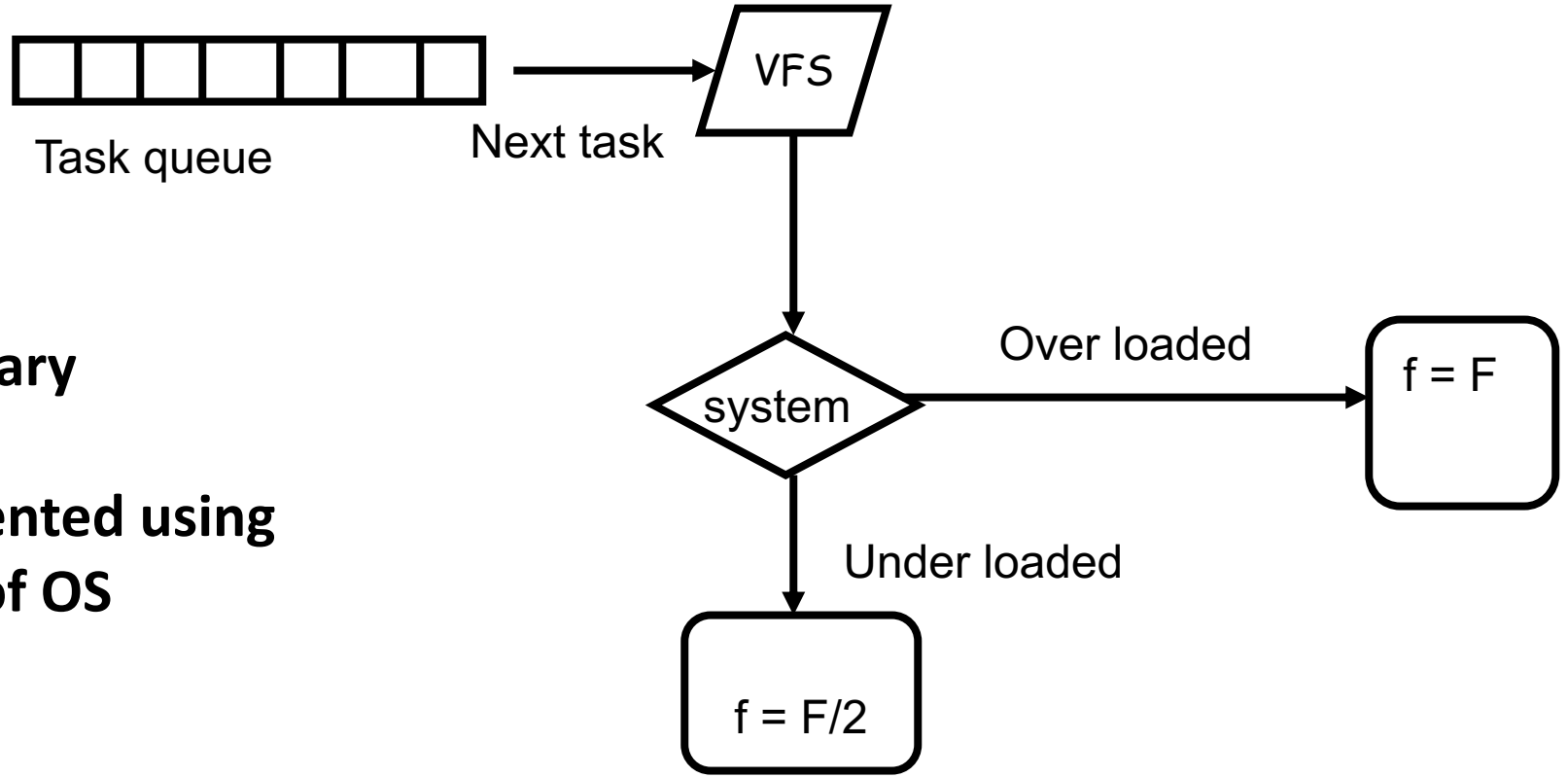


<V, F> pairs for each core



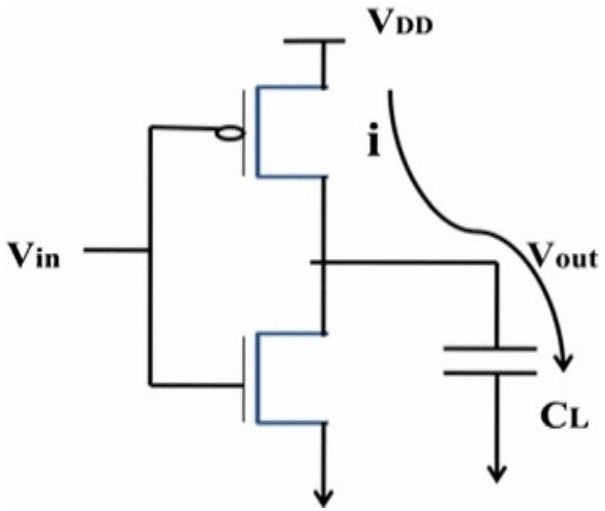
- Our problem:
  - Circuit is given, no changes allowed
  - Only allowed to change V and F values

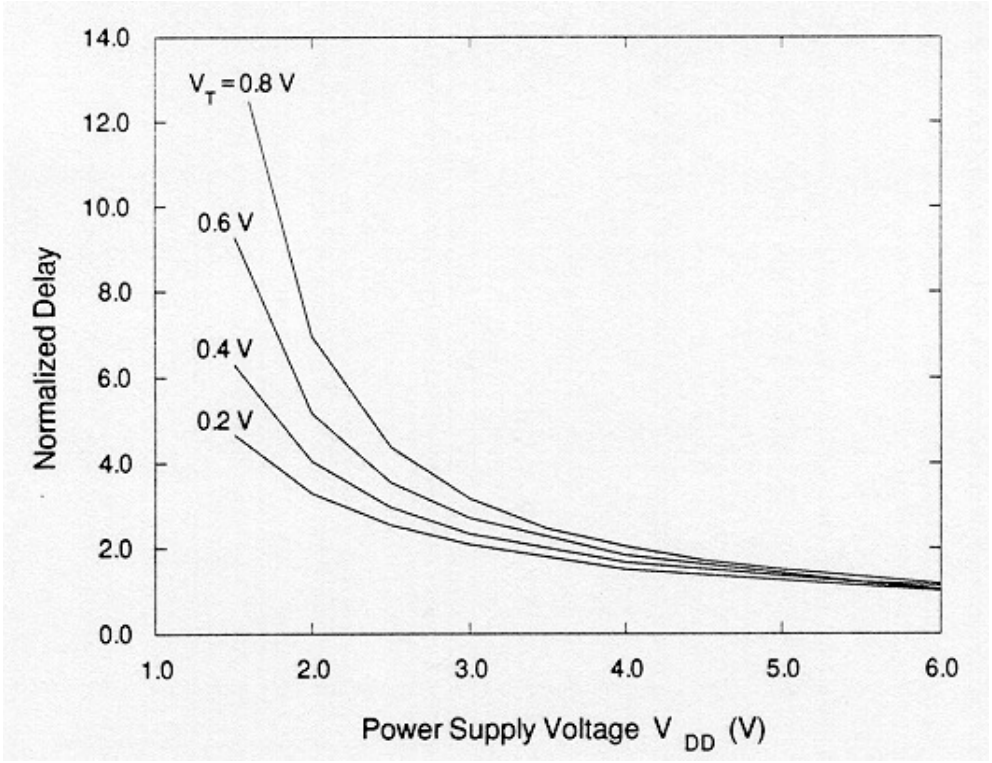
Example of VFS



- **Depending on task, vary voltage/frequency**
- **Entire setup implemented using hardware with help of OS**

Why scale V and F together(1)?



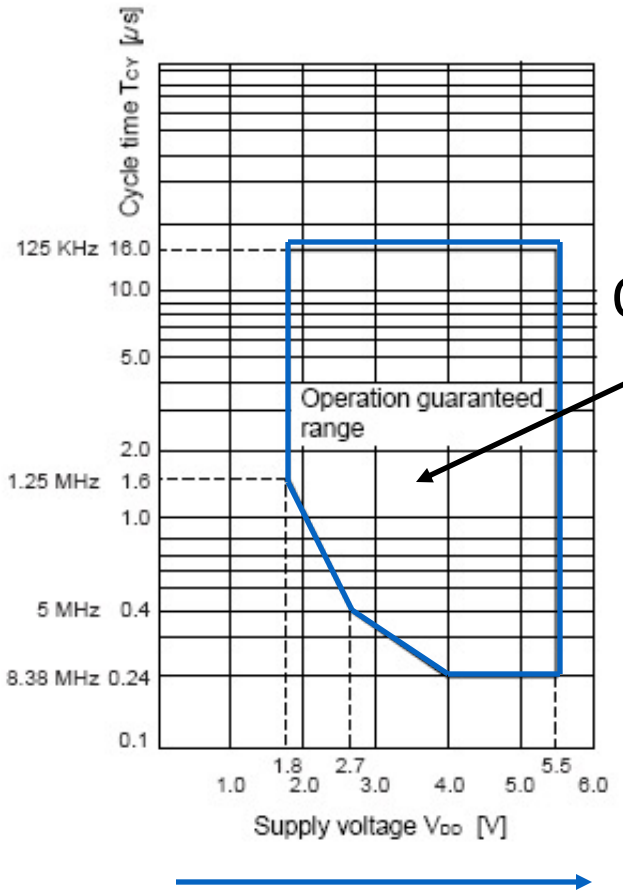


- Reducing V decreases pull-up strength
- Delay of circuit increases automatically
- Clock frequency must be adjusted

Why scale V and F together(2)?

- **Oscillators are not stable at all voltages for a given frequency**

Increasing f



Operation OK

Increasing Vcc

From slides of Jim Peckol, University of Washington

Nexus Processor VFS

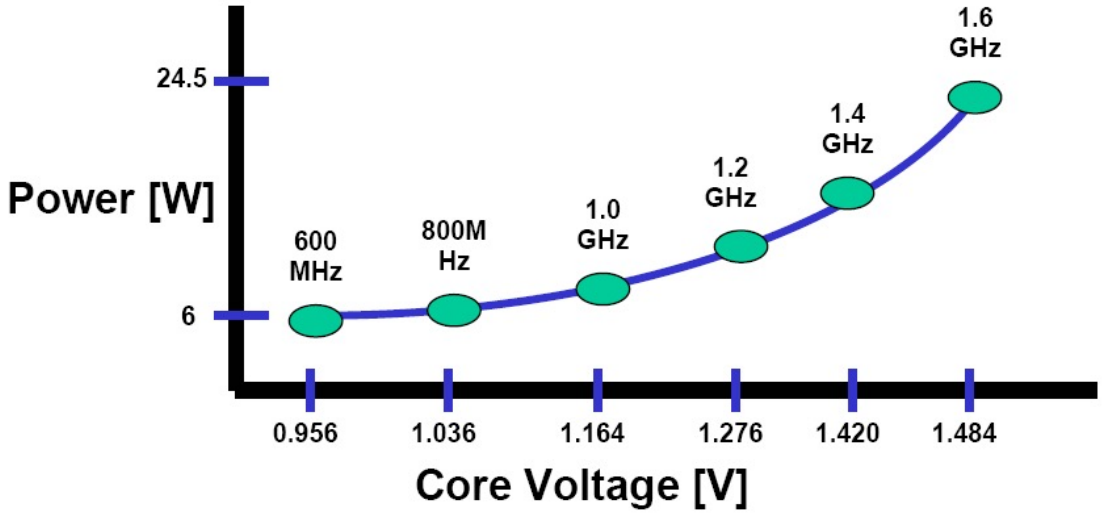
- Running Android 4.1.2 Jelly Bean OS.
- ARM Cortex A8 Hummingbird Processor
  - Supports dynamic frequency scaling from 100Mhz to 1Ghz
  - Supports voltage scaling from 800mV to 1500mV
- Supported frequencies along with their predefined voltage of operation are as follows
  - 100Mhz -> 950mV
  - 200Mhz -> 950mV
  - 400Mhz -> 1050mV
  - 800Mhz -> 1200mV
  - 1000Mhz -> 1250mV

From slides of Manimaran Govindarasu, Iowa State University

Intel Pentium processor

Frequency	Voltage
1.6 GHz (HFM)	1.484 V
1.4 GHz	1.420 V
1.2 GHz	1.276 V
1.0 GHz	1.164 V
800 MHz	1.036 V
600 MHz (LFM)	0.956 V

$P = \alpha C V^2 f$



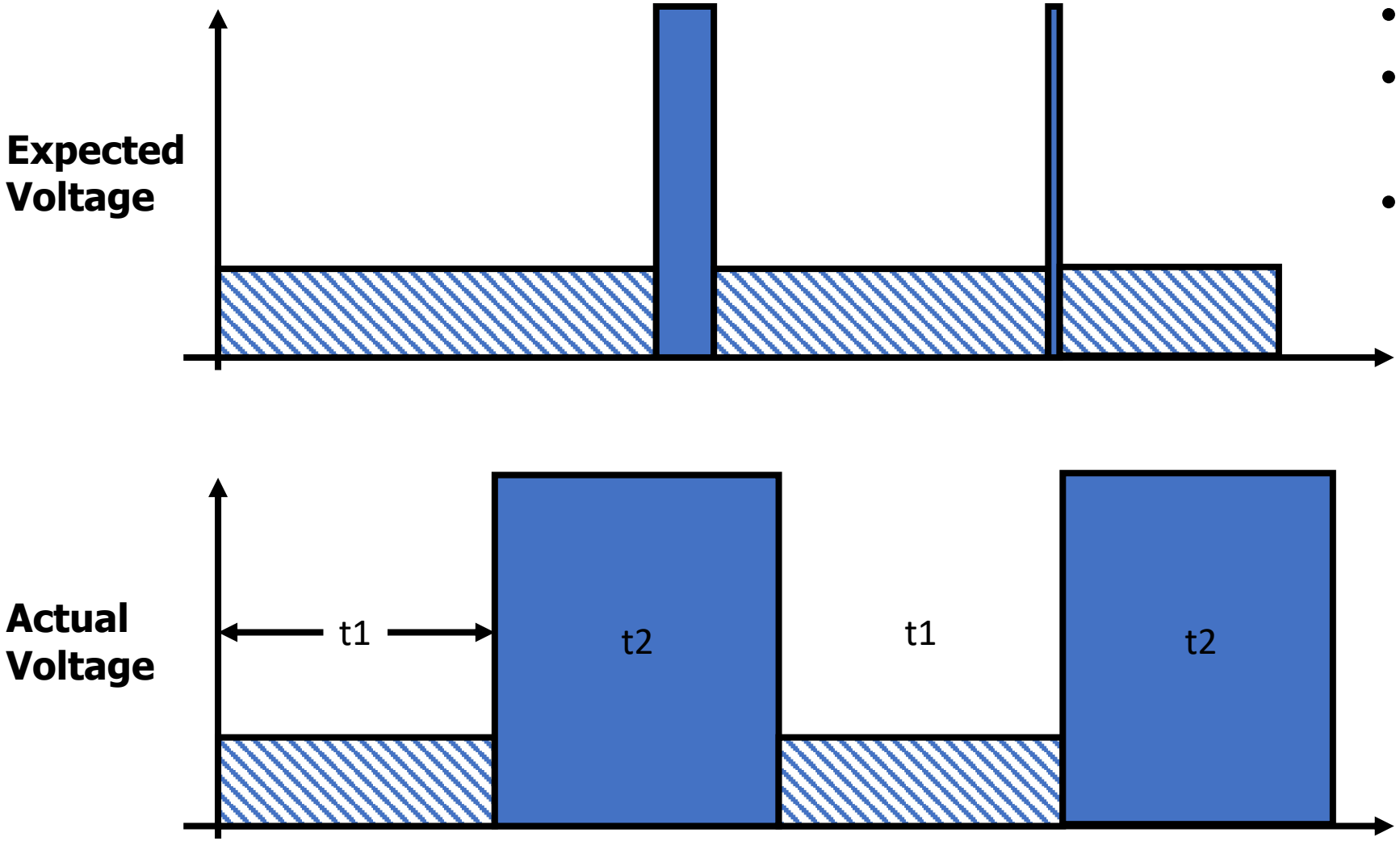
From slides of Jim Peckol, University of Washington

DVFS implementation (software)

- Implemented using Linux kernels
  - Supports multiple CPU governors and schedulers.
- CPU governor
  - Decides how to scale the frequency of the processor based on the workload.
  - Policies: Power save, Conservative, Performance, on demand
- Scheduler
  - Gives tasks access to resources at the time of execution

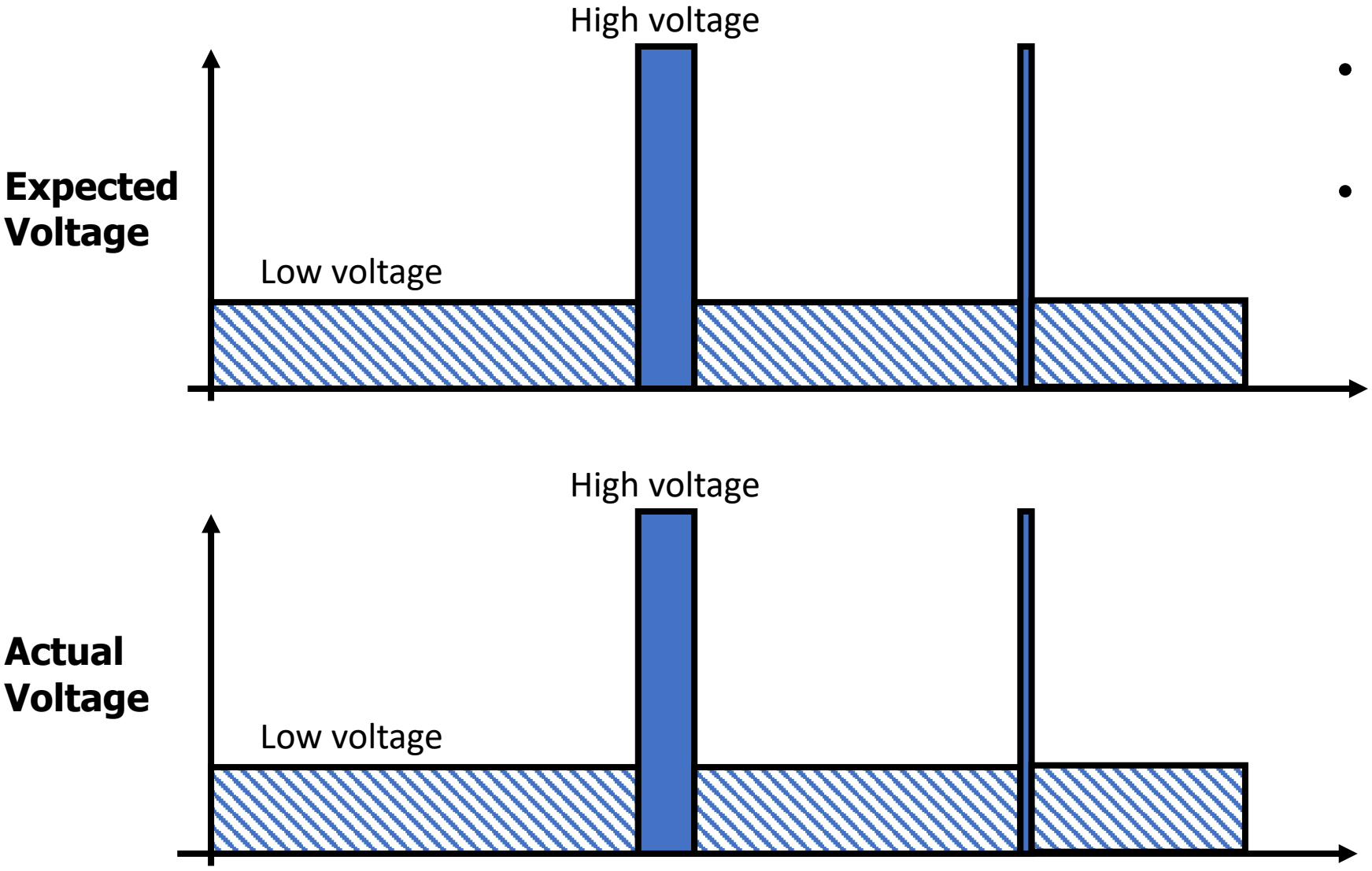
https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt

Static policies for VFS



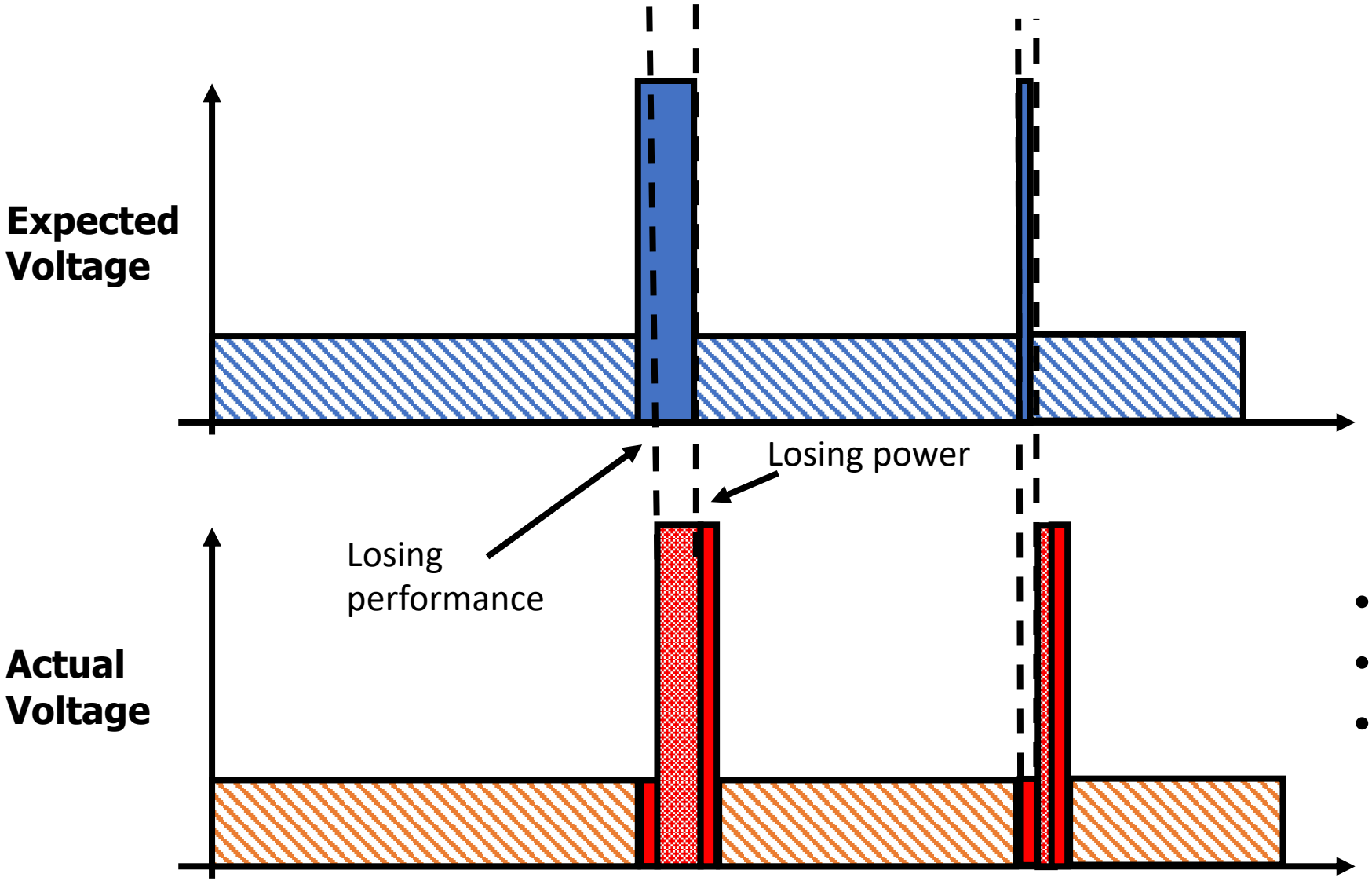
- How to identify changes
- Static policies are easy to implement in hardware
- Sub-optimal power-performance

Dynamic Voltage Frequency Scaling (DVFS)



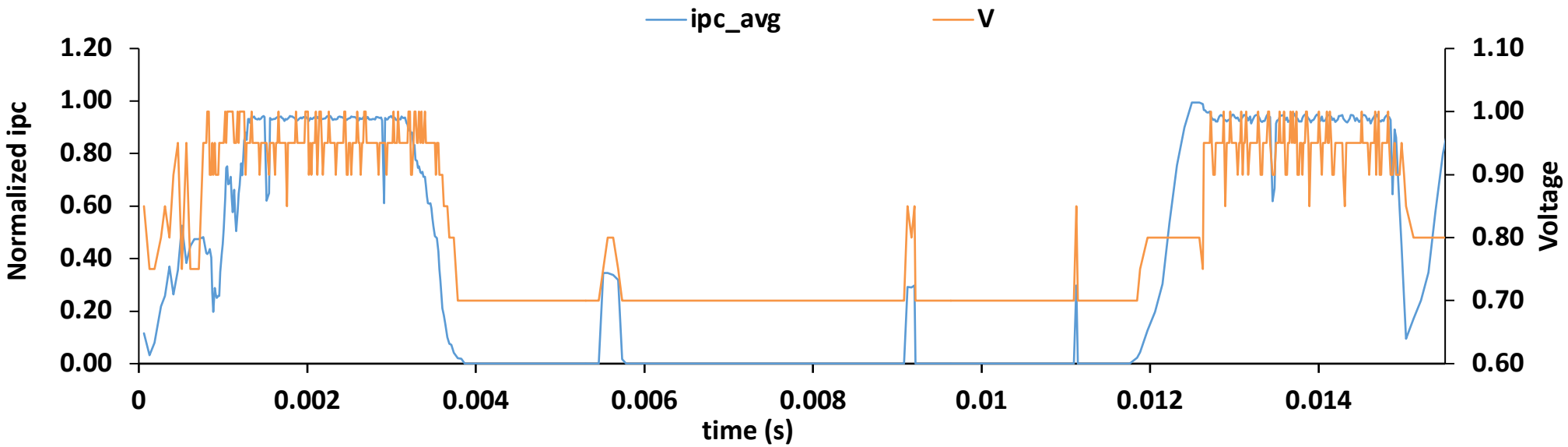
- **Dynamically adapt as per application requirement**
- **Difficult to implement in hardware**

Challenges with Dynamic Voltage Frequency Scaling



- How to identify changes?
- Overhead in switching
- ML prediction?

DVFS example

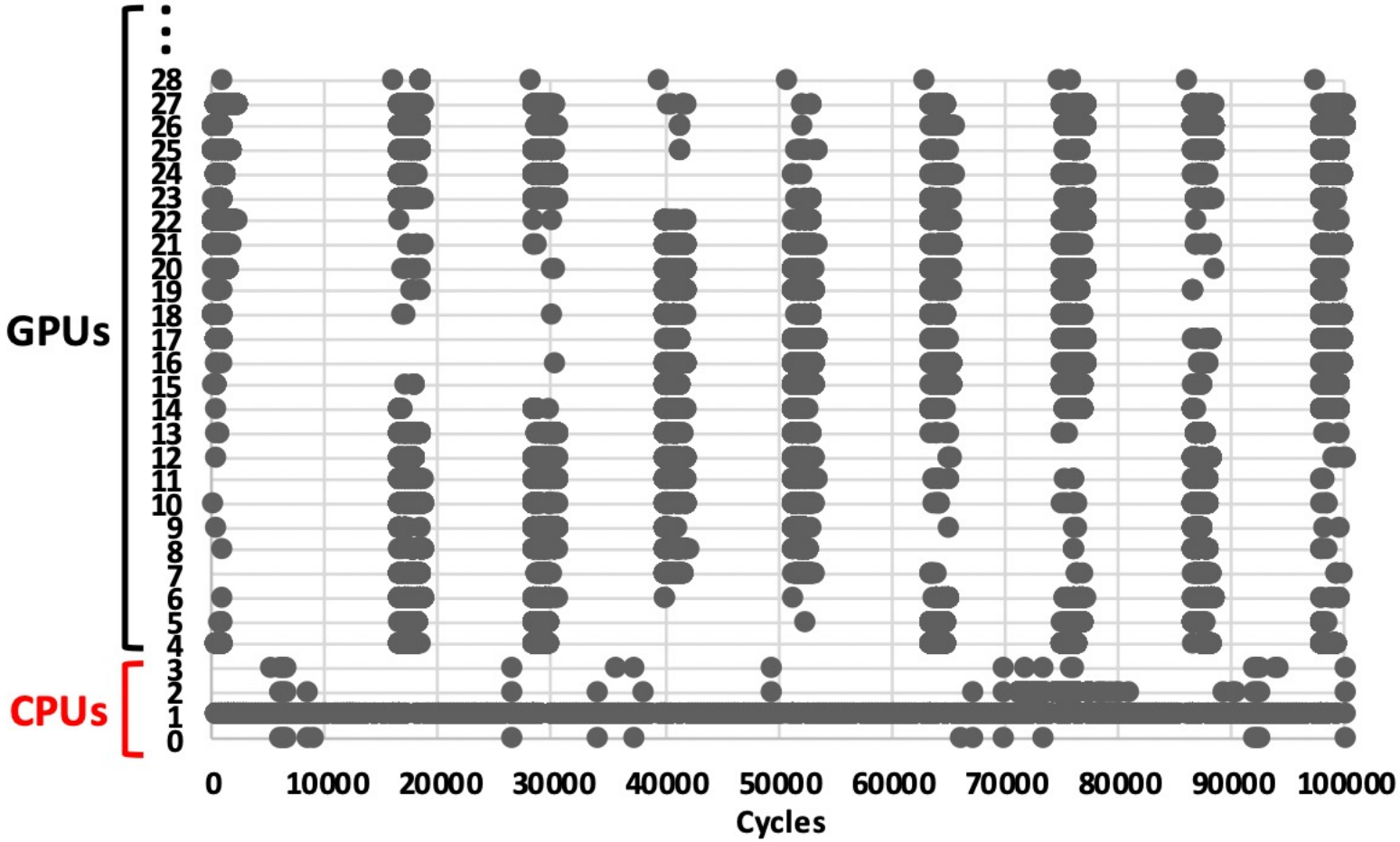


- IPC for FFT with respect to time
- Voltage closely matches the behavior
- ML policy used for prediction

A. Chatterjee, S. Musavvir, R. G. Kim, J. R. Doppa, and P. P. Pande. 2021. Power Management of Monolithic 3D Manycore Chips with Inter-tier Process Variations. J. Emerg. Technol. Comput. Syst. 17, 2, Article 13

Communication by cores

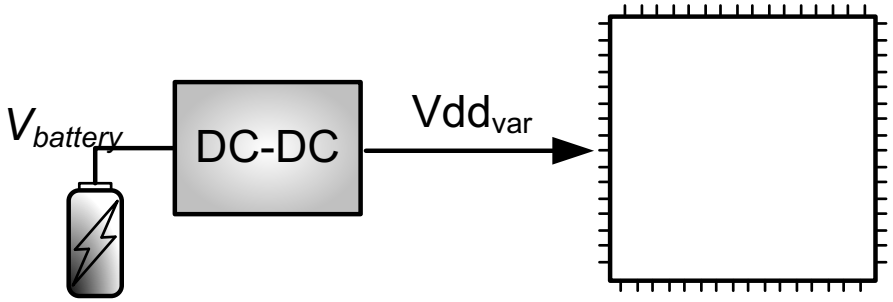
- Similar behavior in communication as well
- Not all cores communicate all the time
- Chance to save power

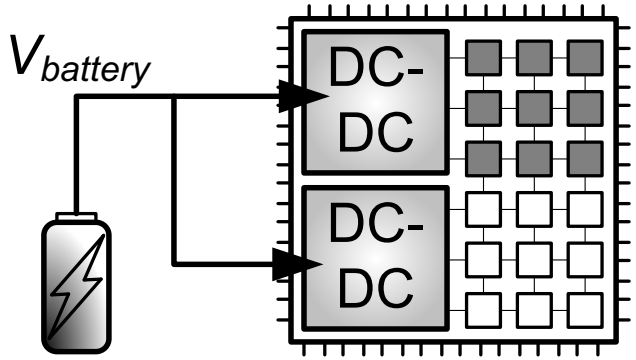


W. Choi et al., "On-Chip Communication Network for Efficient Training of Deep Convolutional Networks on Heterogeneous Manycore Systems," in IEEE Transactions on Computers, vol. 67, no. 5, pp. 672-686, 1 May 2018

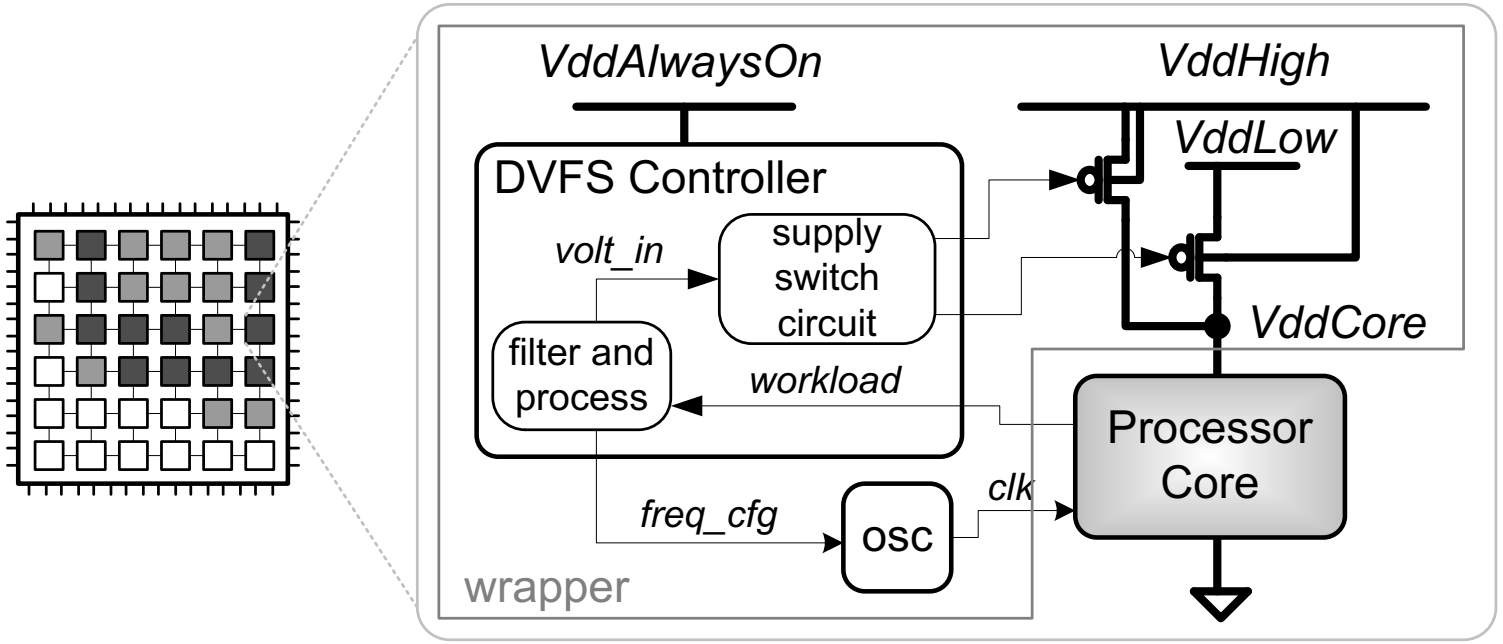
DVFS implementations (hardware)

- Many ways to implement DVFS
- Off-Chip
  - Slower to react
  - Easy to design
- On-Chip
  - Faster to react
  - Difficult to design





DVFS hardware



- **DVFS is enabled by a "DVFS Controller"**
- **DVFS applied on a per-core basis**
- **Often not scalable with number of cores**

From slides of Wayne H. Cheng and Bevan M. Baas, UC Irvine

The controller

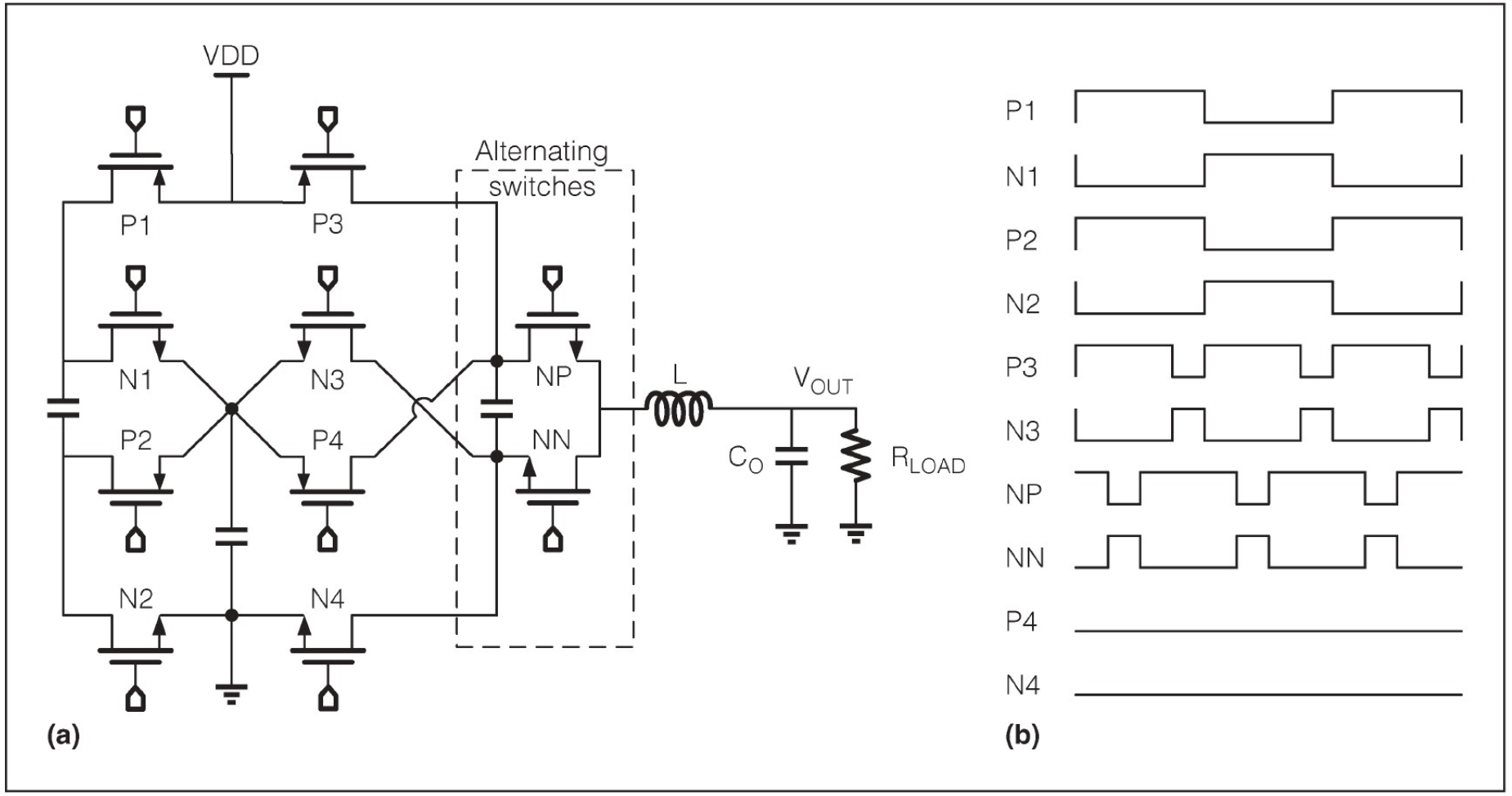
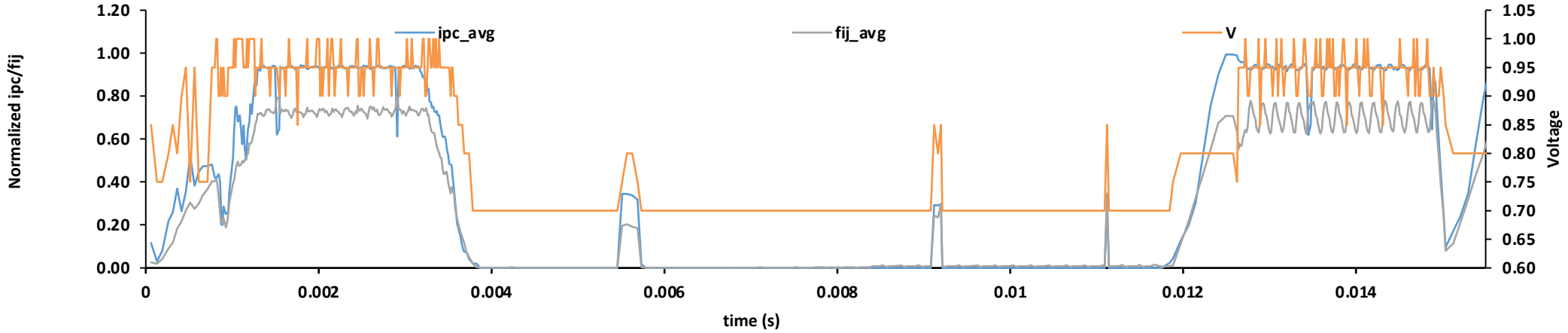


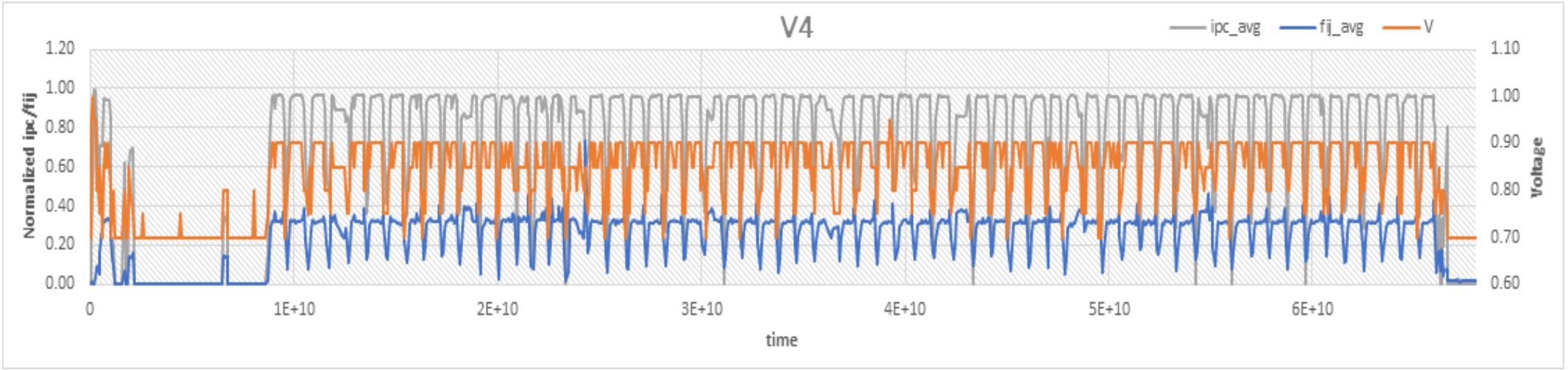
Figure 1. (a) Power stage of hybrid SISC regulator; VDD = 2 V, L = 500 pH, C_O = 2 nF (b) Switching patterns of the switches when V_{OUT} is between VDD and VDD/2, and P4/N4 pair is disabled.

J. Murray, N. Tang, P. P. Pande, D. Heo and B. A. Shirazi, "DVFS Pruning for Wireless NoC Architectures," in IEEE Design & Test, vol. 32, no. 2, pp. 29-38, April 2015

DVFS in manycore



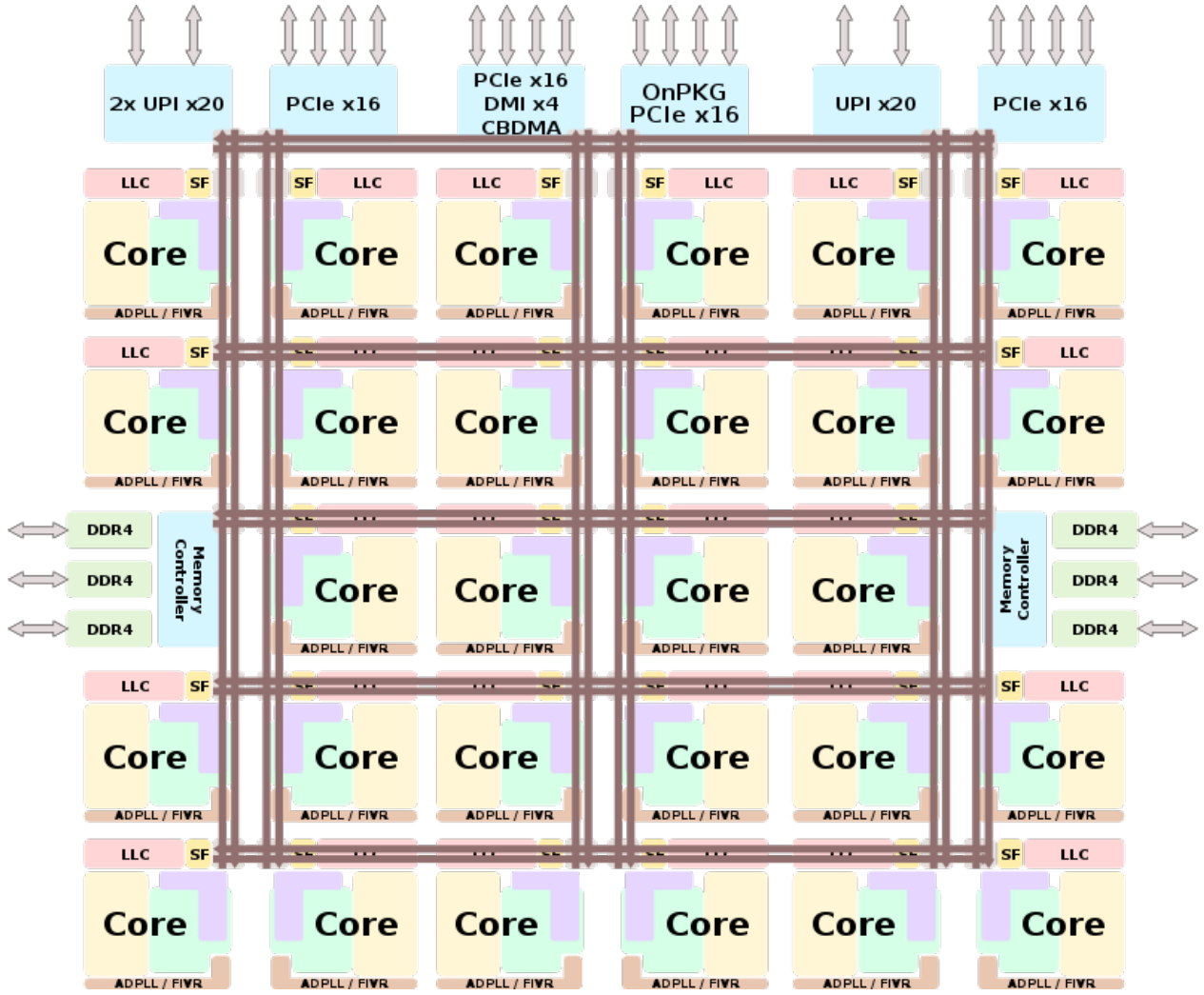
FFT



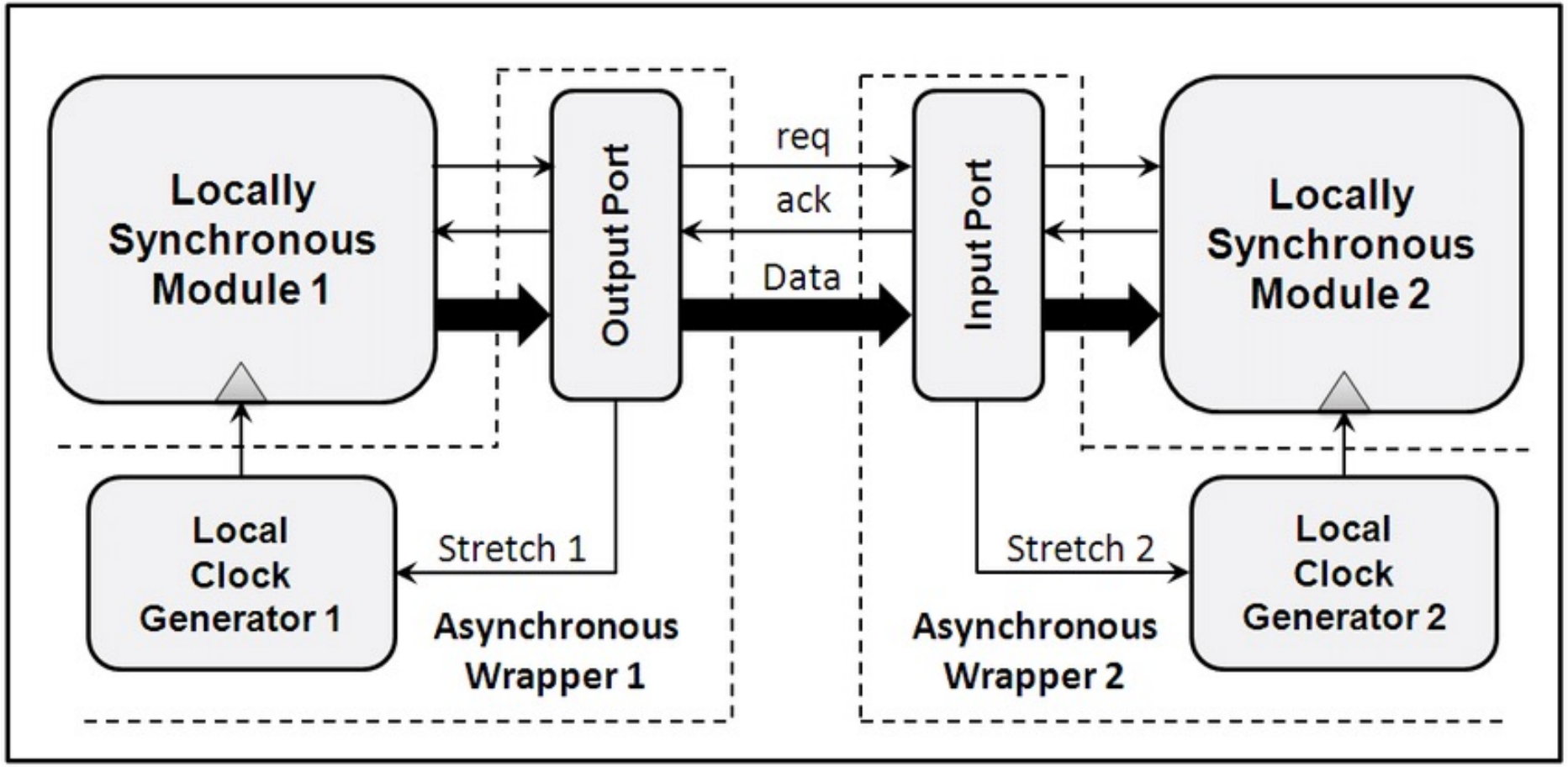
Canneal

DVFS challenges

- One controller per core
- Multiple V/F configurations
- Communication is challenge



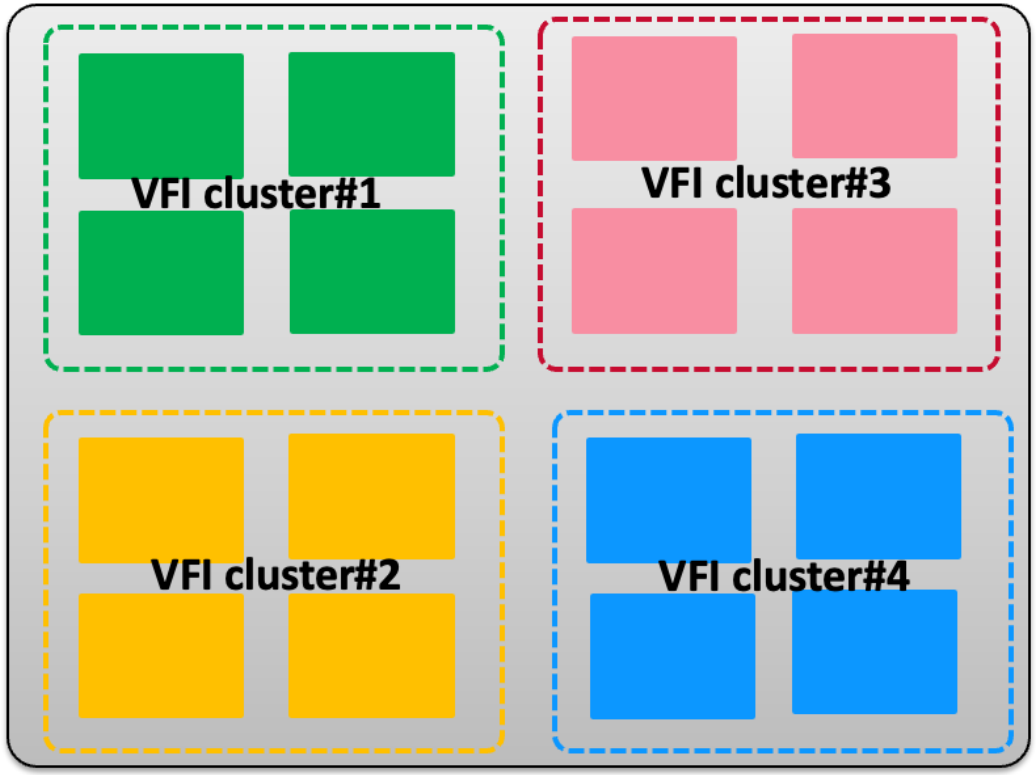
Communication challenge in DVFS



- Also known as Globally asynchronous locally synchronous (GALS) system
- Communication is based on acknowledgements

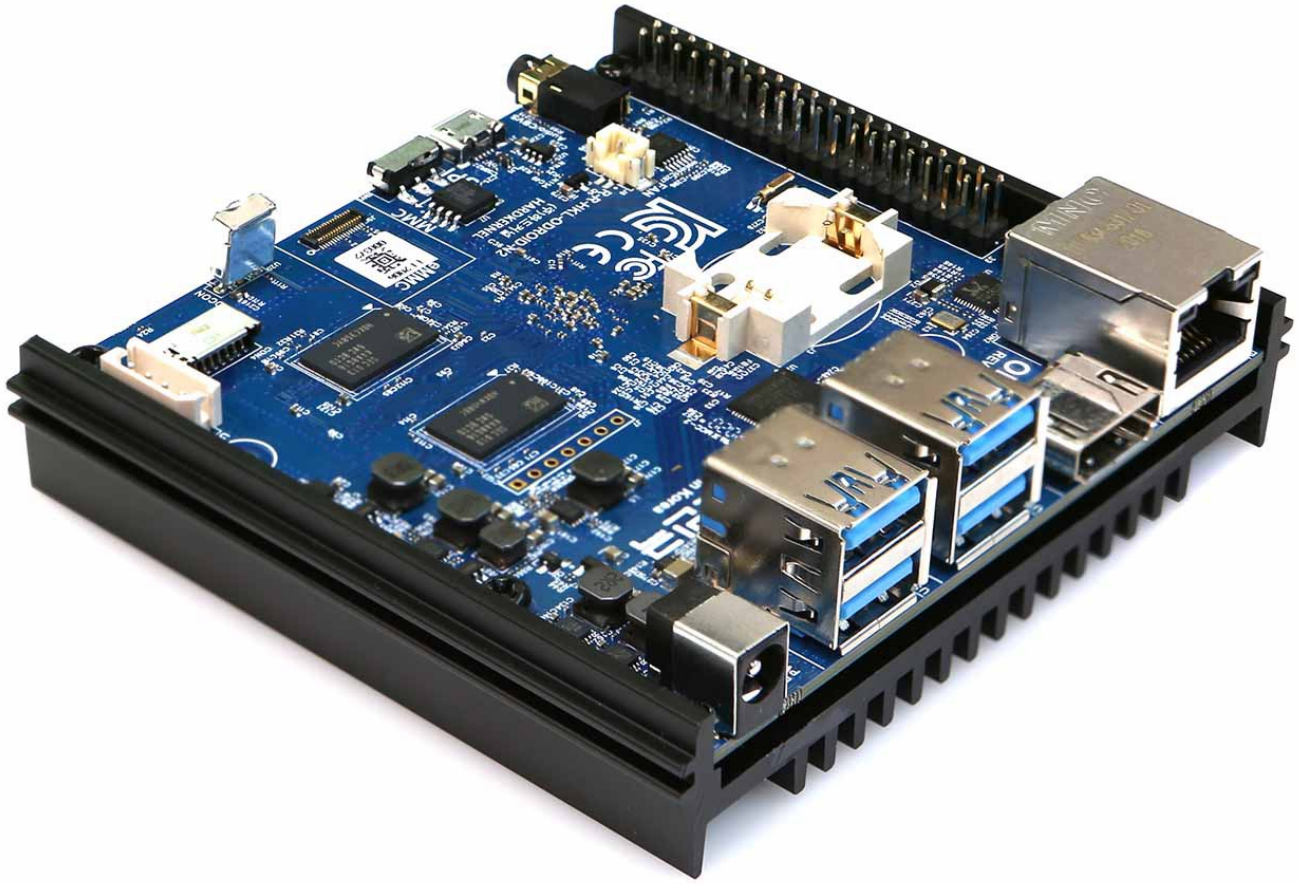
Voltage Frequency Islands (VFI)

- Birds of a feather flock together
- Have one controller per cluster
- Cores belonging to same cluster have same voltage and frequency
- ARM board (Big-Little clusters)



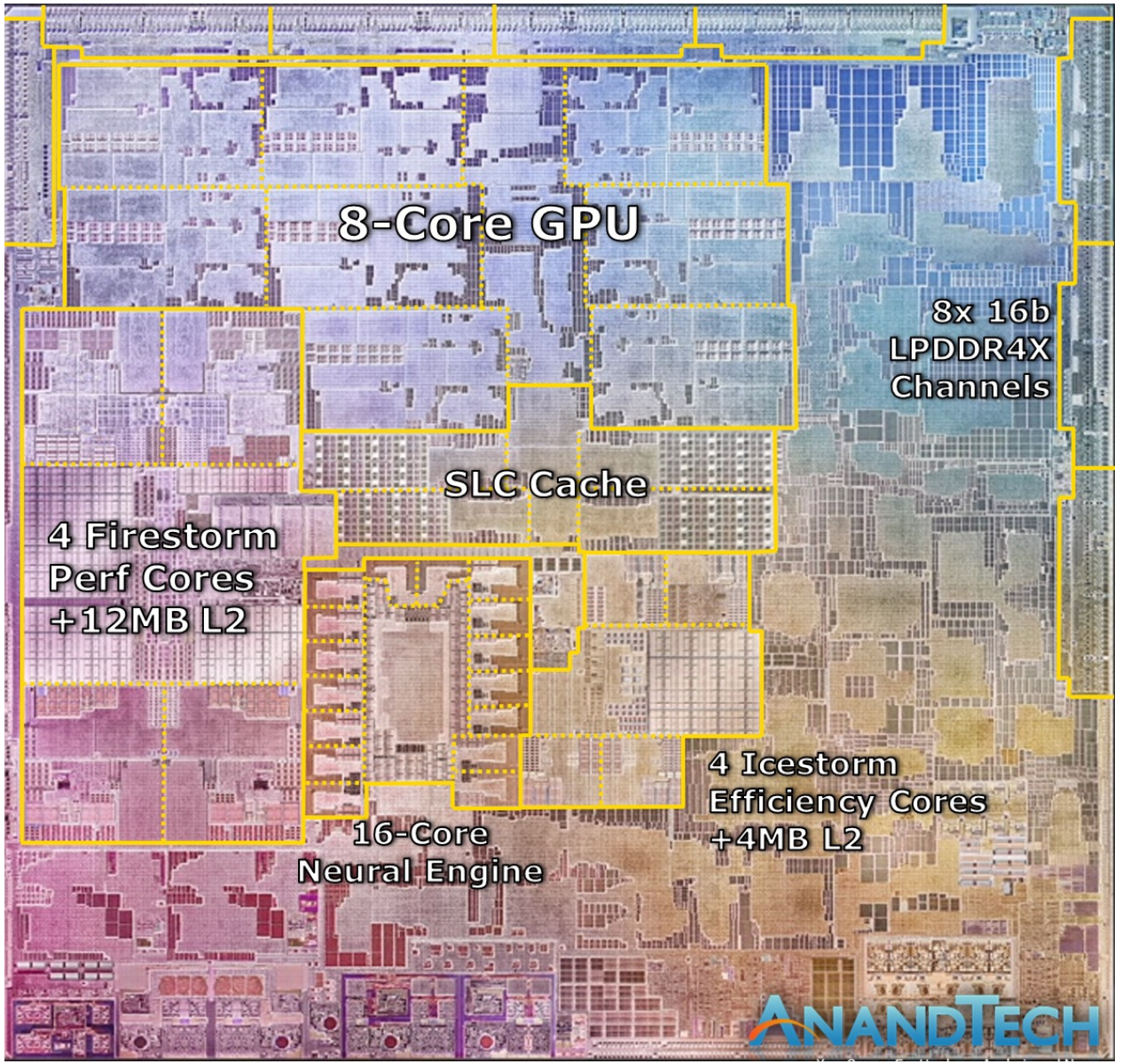
ARM board (Odriod xu4)

- ARM board
- Two computing clusters
- Big-Little clusters
- Performance → Big cluster
- Power →Little cluster
- Map tasks according to requirement



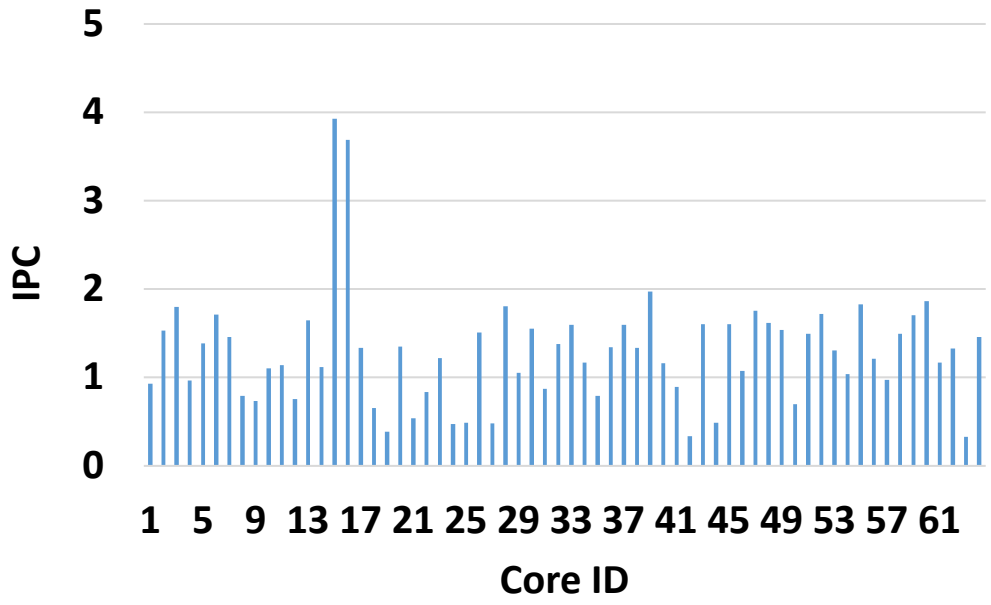
Apple M1 chip

- Two types of CPUs
  - Performance cores
  - Efficiency cores

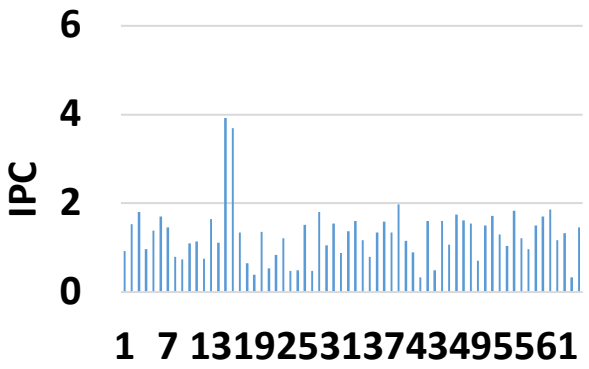


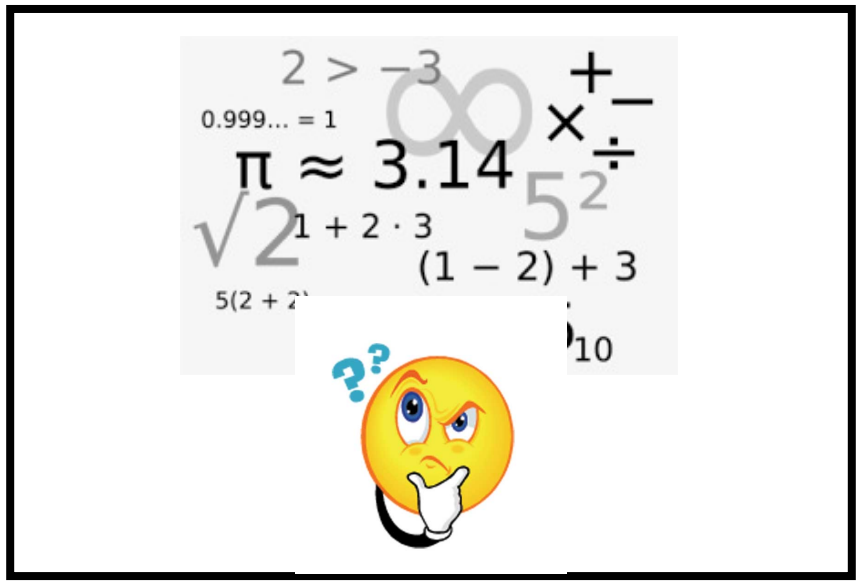
Manycore application characteristics

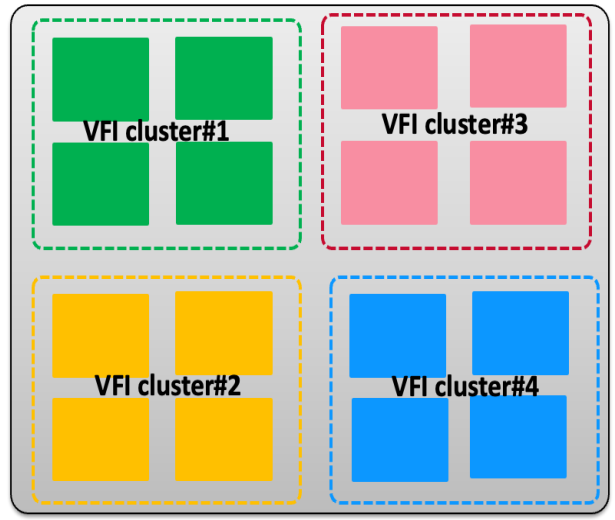
- Per-core DVFS is costly
- Many cores have similar IPC
- Create groups of similar cores, a.k.a., islands
- One controller per cluster



Voltage Frequency Island







- A group of cores have similar behavior
- <V,F> pairs for each group instead of each core
- Fewer DVFS controllers
- Easier communication

Overall VFI methodology

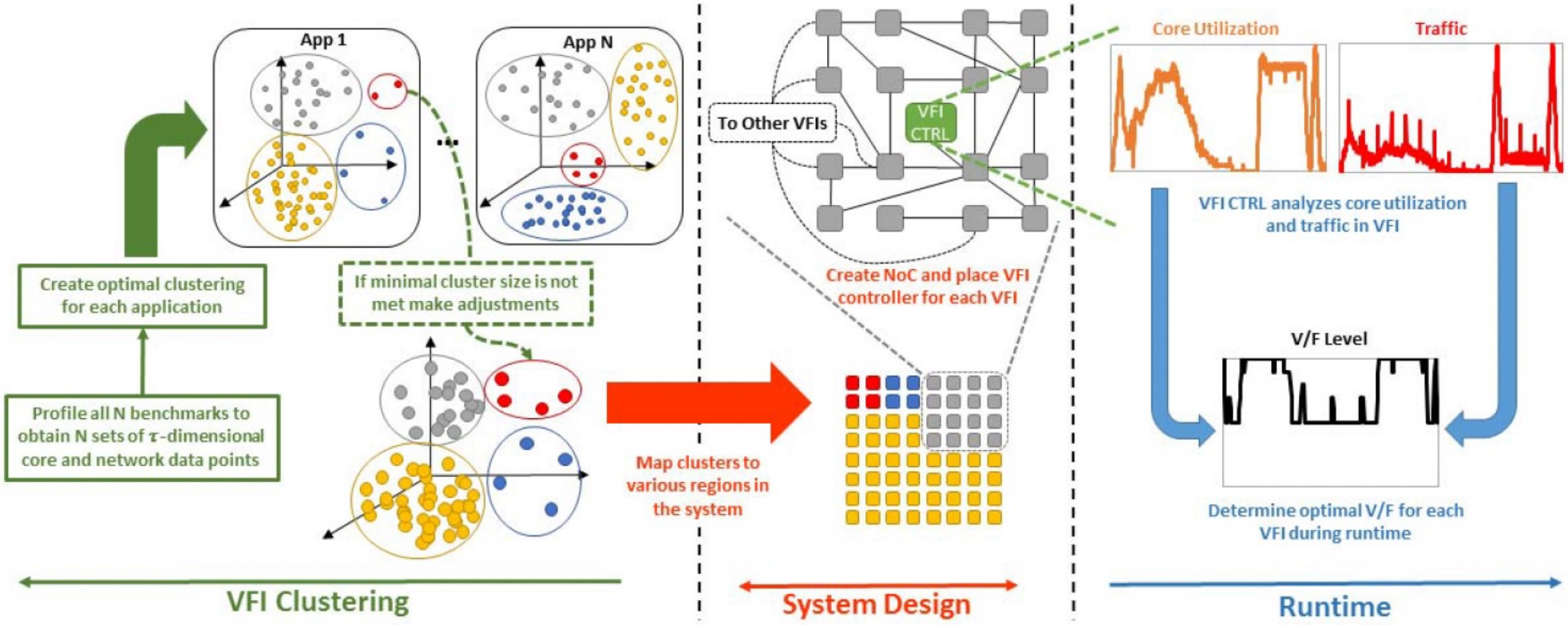
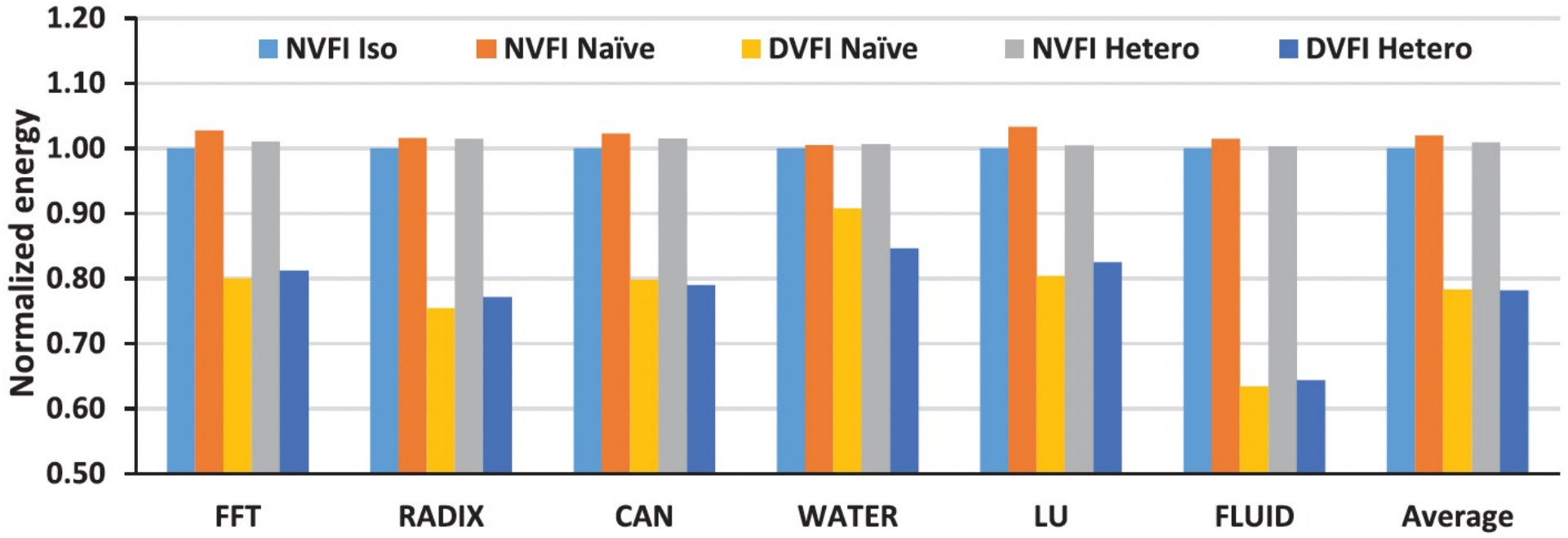


Fig. 1. Illustration of the proposed VFI-partitioned multicore codesign methodology. During the VFI clustering phase, each application is profiled in order to obtain key time-varying core and network statistics. These statistics are used to find the optimal clustering, within particular size constraints, for each application. During system design, these clusters are then mapped to physical cores. The NoC and VFI controller (VFI CTRL), the dedicated hardware block that dynamically tunes the V/F of the VFI, are then placed to tailor to the traffic and cluster characteristics of each application. During runtime, the VFI CTRL obtains the core and traffic data in order to determine optimal V/F levels for the VFI.

R. G. Kim, W. Choi, Z. Chen, P. P. Pande, D. Marculescu and R. Marculescu, "Wireless NoC and Dynamic VFI Codesign: Energy Efficiency Without Performance Penalty," in IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 24, no. 7, pp. 2488-2501, July 2016.

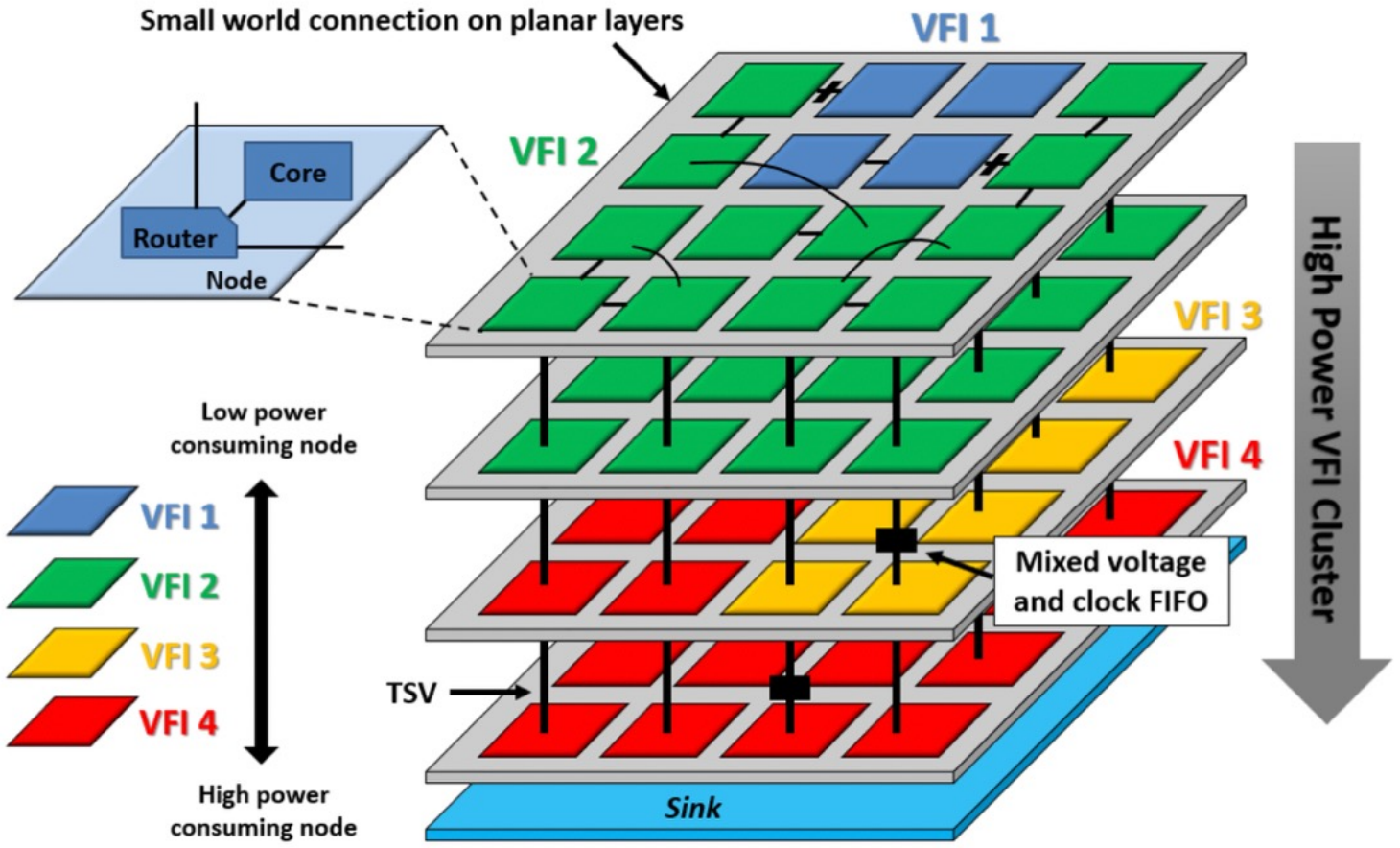
Effectiveness of VFI



- **Average energy improvement for the Naïve M3D systems is 21.68%**
- **Energy improvement is 21.87% for the Hetero M3D systems**

A. Chatterjee, S. Musavvir, R. G. Kim, J. R. Doppa, and P. P. Pande. 2021. Power Management of Monolithic 3D Manycore Chips with Inter-tier Process Variations. In J. Emerg. Technol. Comput. Syst., 17, 2, Article 13 (April 2021), 19 pages.

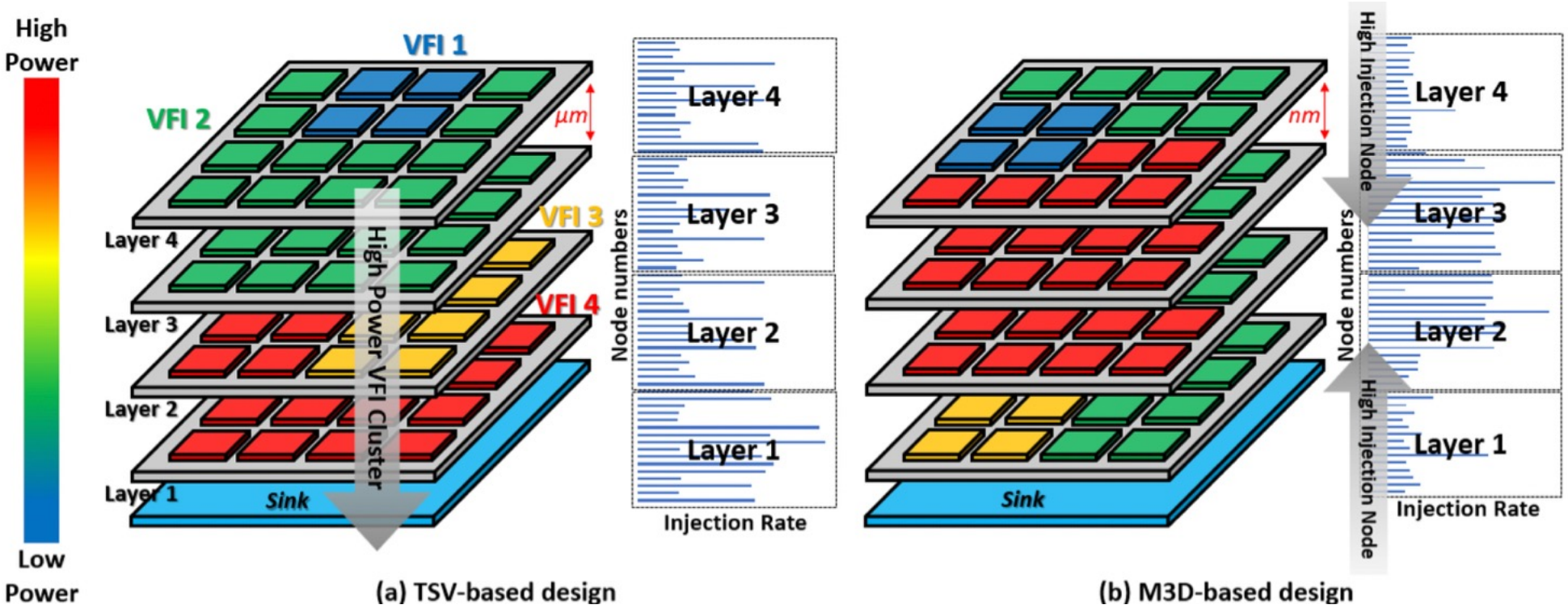
VFI in 3D



- **VFI can be done with 3D architectures as well**
- **Higher V/F cores near the sink**

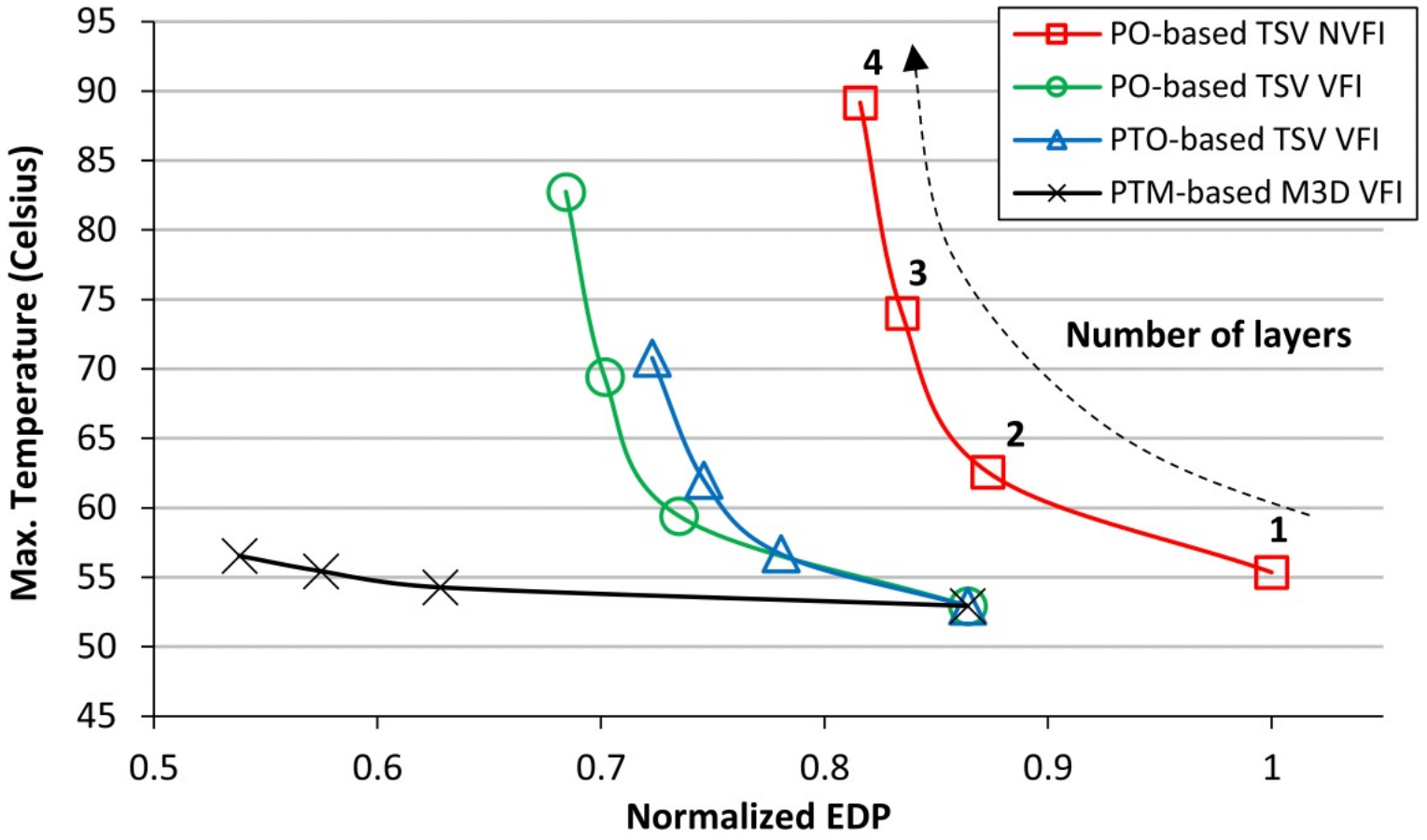
Dongjin Lee, Sourav Das, Janardhan Rao Doppa, Partha Pratim Pande, and Krishnendu Chakrabarty. 2018. Performance and Thermal Tradeoffs for Energy-Efficient Monolithic 3D Network-on-Chip. ACM Trans. Des. Autom. Electron. Syst. 23, 5, Article 60 (October 2018), 25 pages.

VFI: TSV vs M3D



- VFI for TSV and M3D are different
- For M3D, higher V/F cores not near sink

VFI outcome: TSV vs M3D



- **M3D architectures have better power-performance trade-offs**
- **Pareto front is closer to origin**

Experimental setup

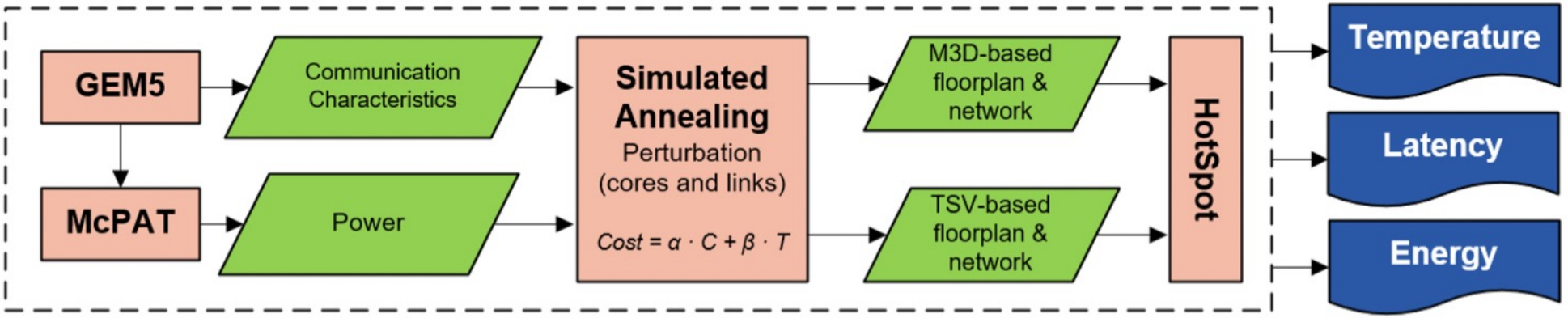


Fig. 6. Experimental set up for establishing the performance and thermal tradeoff for TSV- and M3D-based NoC architectures.

- **Use open-source simulators to get the data**