ΕΠΛ 605: Προχωρημένη Αρχιτεκτονική Υπολογιστών

Presentation of

UniServer Horizon 2020 European project findings:

X-Gene server chips, voltage-noise characterization, high-bandwidth voltage measurements, dI/dt viruses, V_{MIN} characterization

UniServer Project Overview

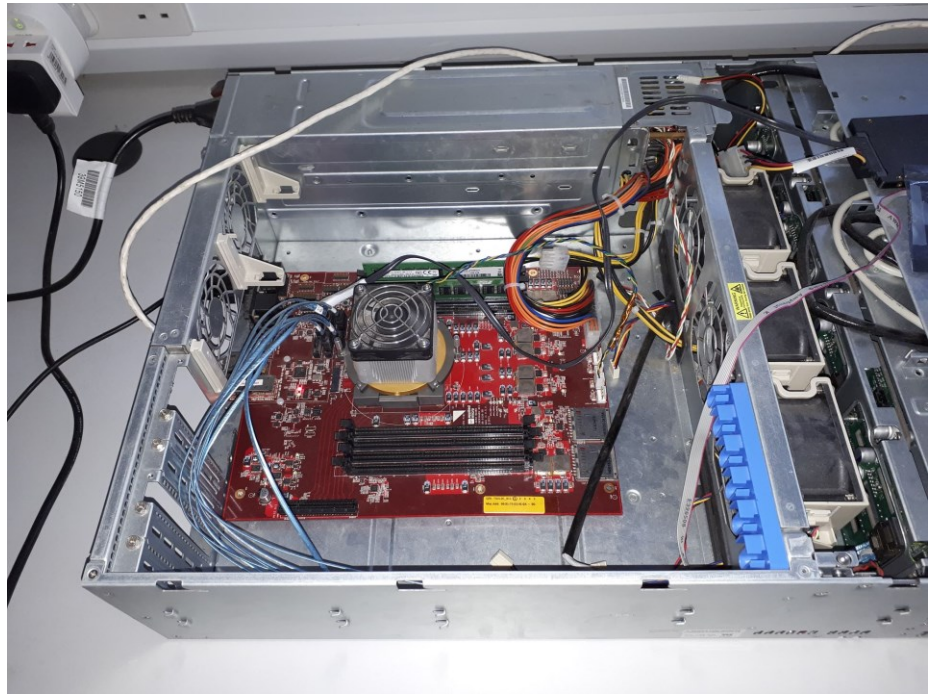
- 3.5 years major research project funded by European Community's Horizon 2020 research program.
- Project budget 4.8 million
- Started in February 2016 to finish in July 2019
- Aims to develop universal system and software architecture for servers targeting cloud and edge computing market
- Key principle behind Uniserver approach is exposing the hardware intrinsic variations by pushing the operating voltage, frequency, refresh rates points beyond the pessimistic nominal values

UniServer Partners

Participant No	Participant organisation name	Part. Short name	Country
1. (Coordinator)	The Queen's University of Belfast	QUB	United Kingdom
2.	University of Cyprus	UCY	Cyprus
3.	University of Athens	UOA	Greece
4.	Applied Micro Circuits Corporation Deutschland GmbH	APM	Germany
5.	ARM Holdings	ARM	United Kingdom
6.	IBM Ireland Ltd	IBM	Ireland
7.	University of Thessaly	UTH	Greece
8.	Worldsensing	WSE	Spain
9.	Meritorius Audit Ltd	MER	Cyprus
10.	Sparsity	SPA	Spain

- Applied Micro provides the hardware, the X-Gene2 and X-Gene3 state-of-the-art ARM 64bit server CPUs
- QUB, UCY, UOA, ARM lead the hardware characterization effort that will reveal the pessimistic operating points
- UTH and IBM lead the effort of developing fault-tolerant hypervisor and resource managers (e.g. open stack)
- WSE, MER and SPA provide the application where the Uniserver software hardware ecosystem will be evaluated on

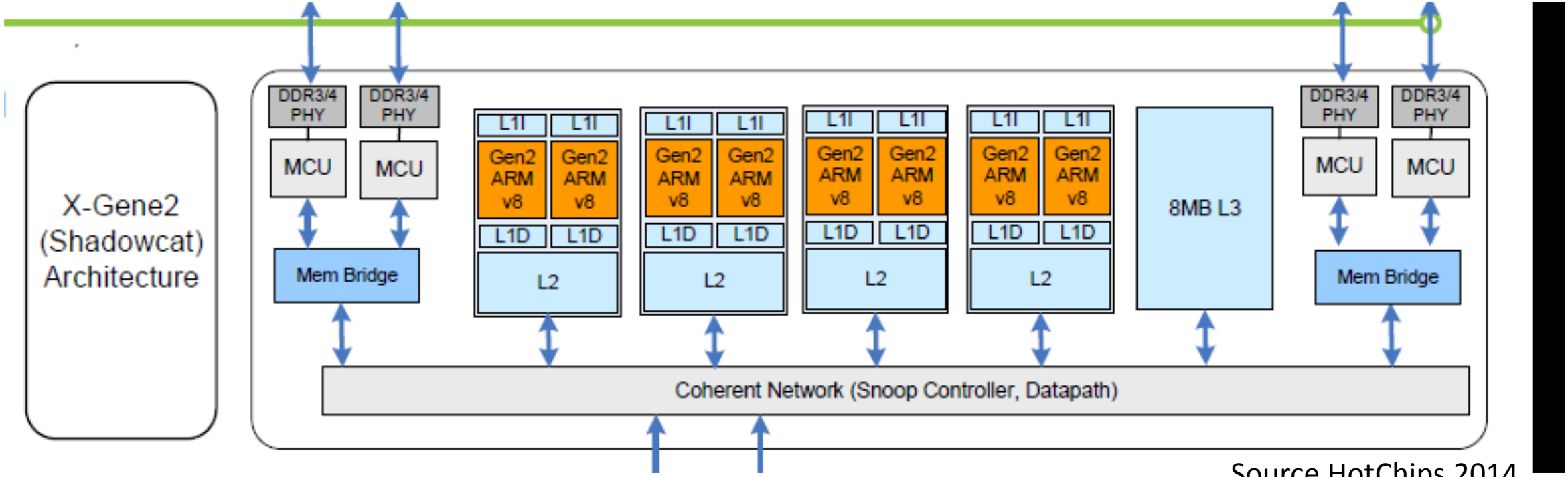
UniServer Hardware



- X-Gene2 server board
- 8 cores @ 2.4GHz, 0.98V at 28nm
- 8MB LLC cache
- 32 GB DDR3

- X-Gene3 server board
- 32 cores @ 3GHz, 0.87V at 16nm
- 32MB LLC cache
- 128GB DDR4

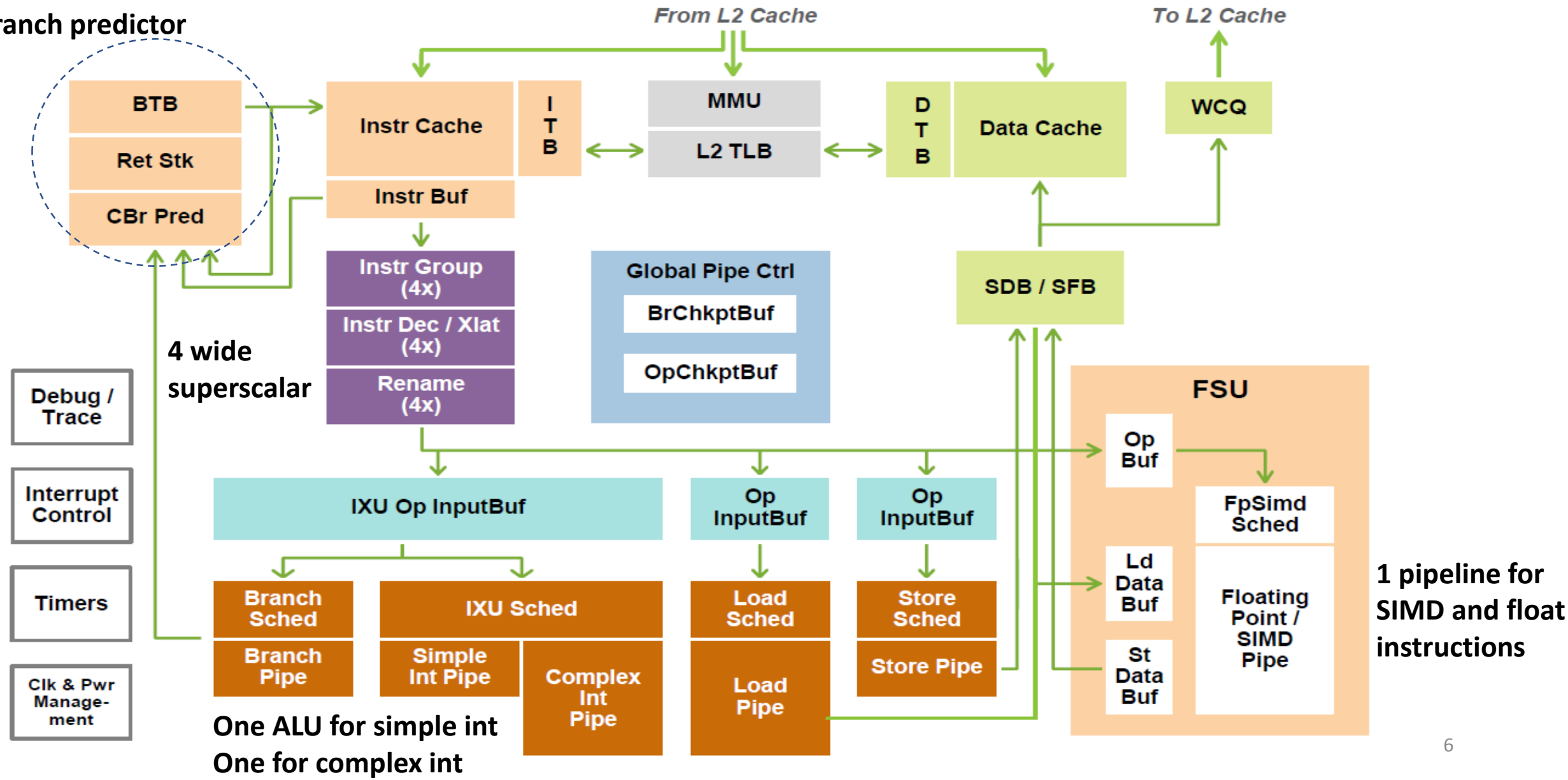
X-Gene 2/3 chip layout



Source HotChips 2014

- Cores are packed into PMD (processor modules)
- Each PMD has two cores and one shared among the two cores L2 cache (256KB)
- Each core has private 32KB L1I and L1D
- L1 cache is write through to L2. *Questions: What is the optimal allocation strategy of threads to cores for performance? Is this like SMT?*
- 8MB L3 is shared among all cores

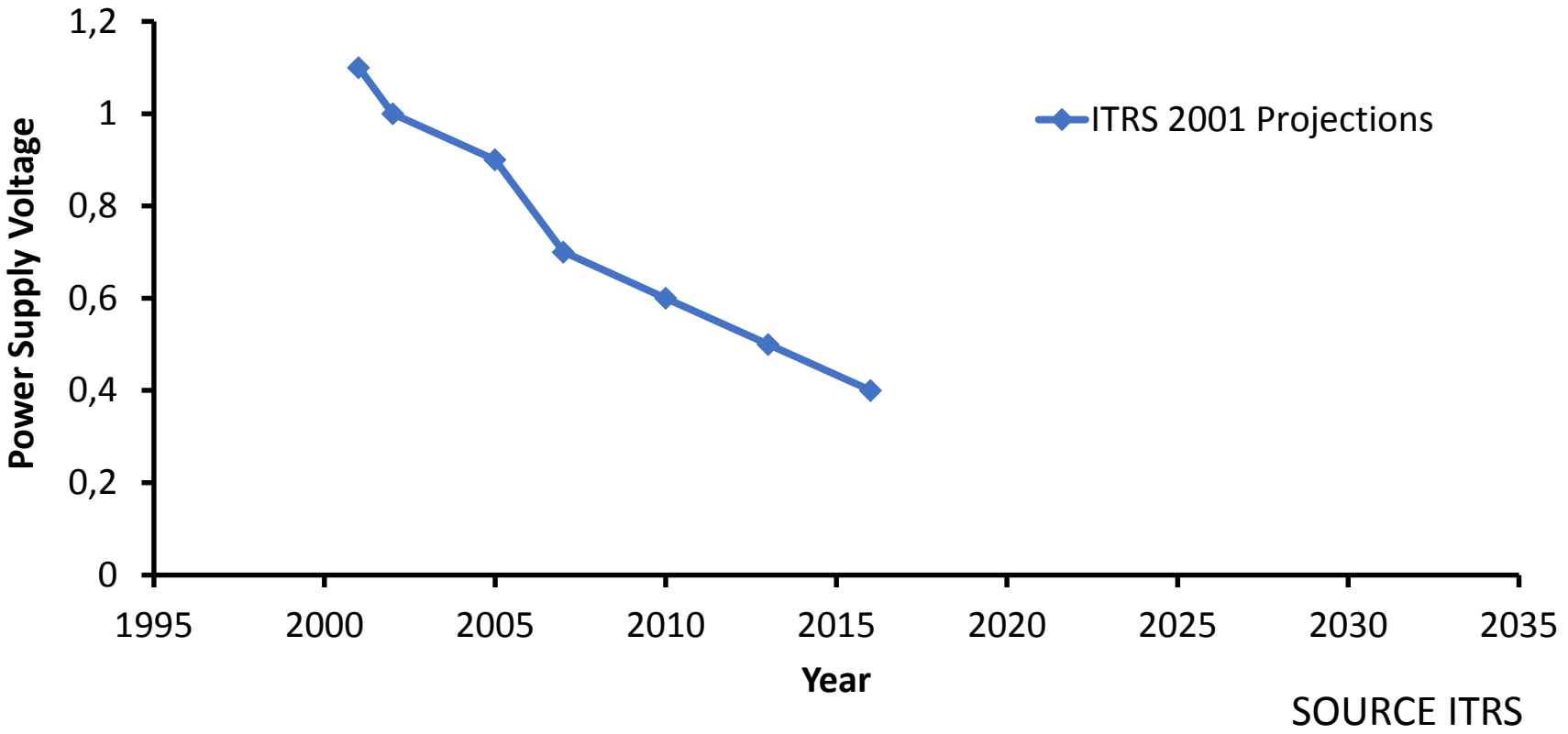
X-Gene2 uArch block diagram



X-Gene Voltage knobs

- X-Gene2/3 servers offer 3 voltage domains
  - CPU, LLC, DRAM
- Hence we can optimize the voltage for CPU, LLC and DRAM
- Moreover we can optimize the **DRAM refresh-rate**

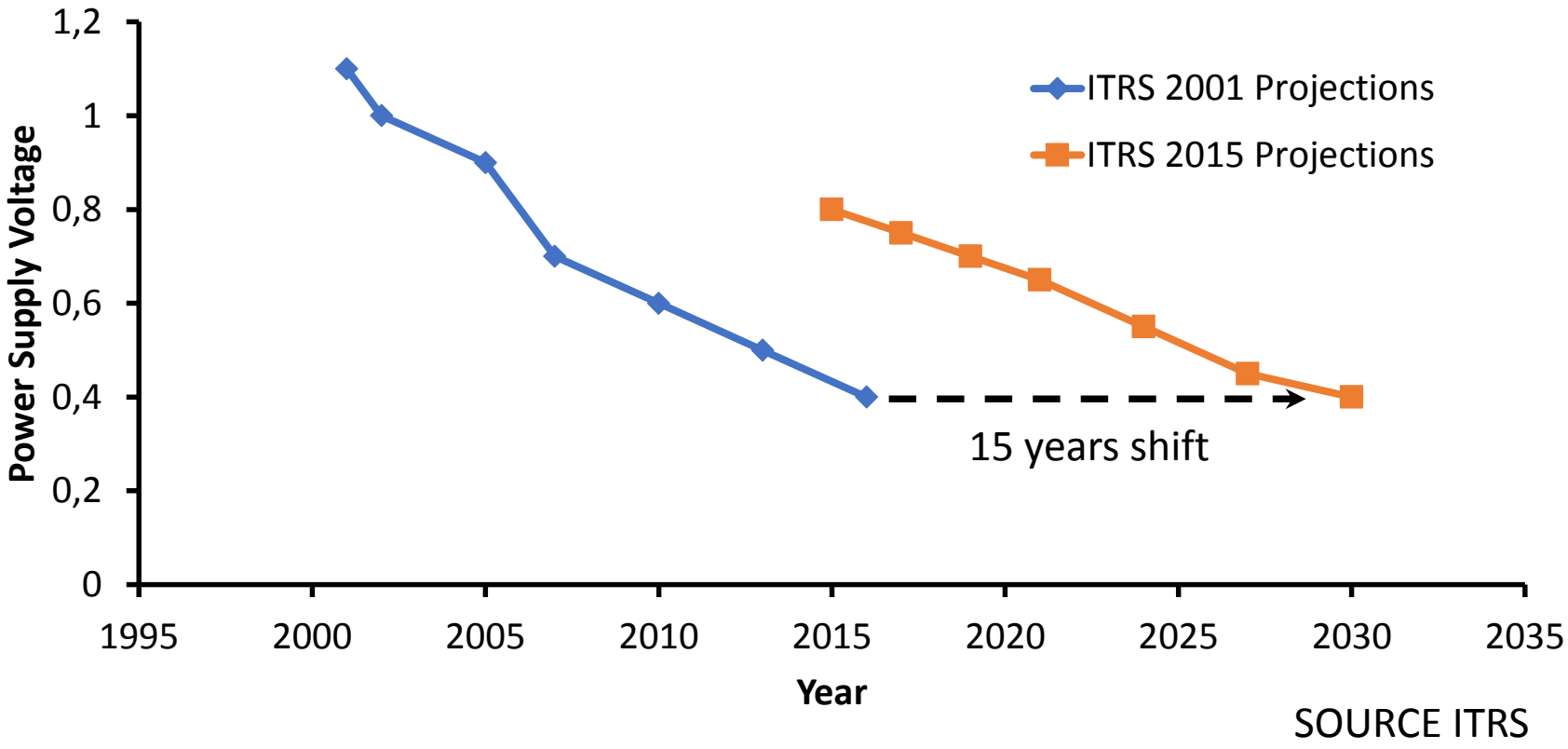
UniServer Motivation - Dennard scaling



SOURCE ITRS

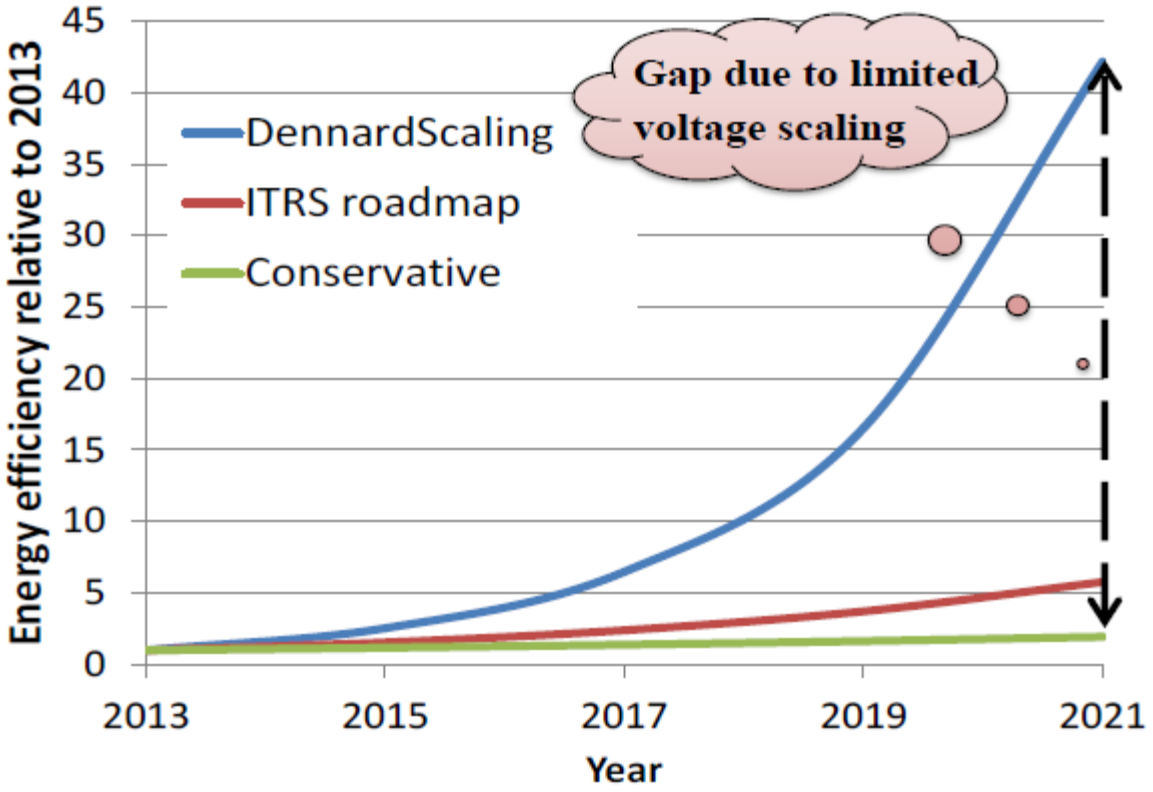
Voltage used to scale linearly with transistor dimensions following Dennard's Rule

End of Dennard Scaling



- Limited voltage scaling since ~2005
- **ITRS 2001 projections fell significantly off**

Limited Energy Efficiency Gains



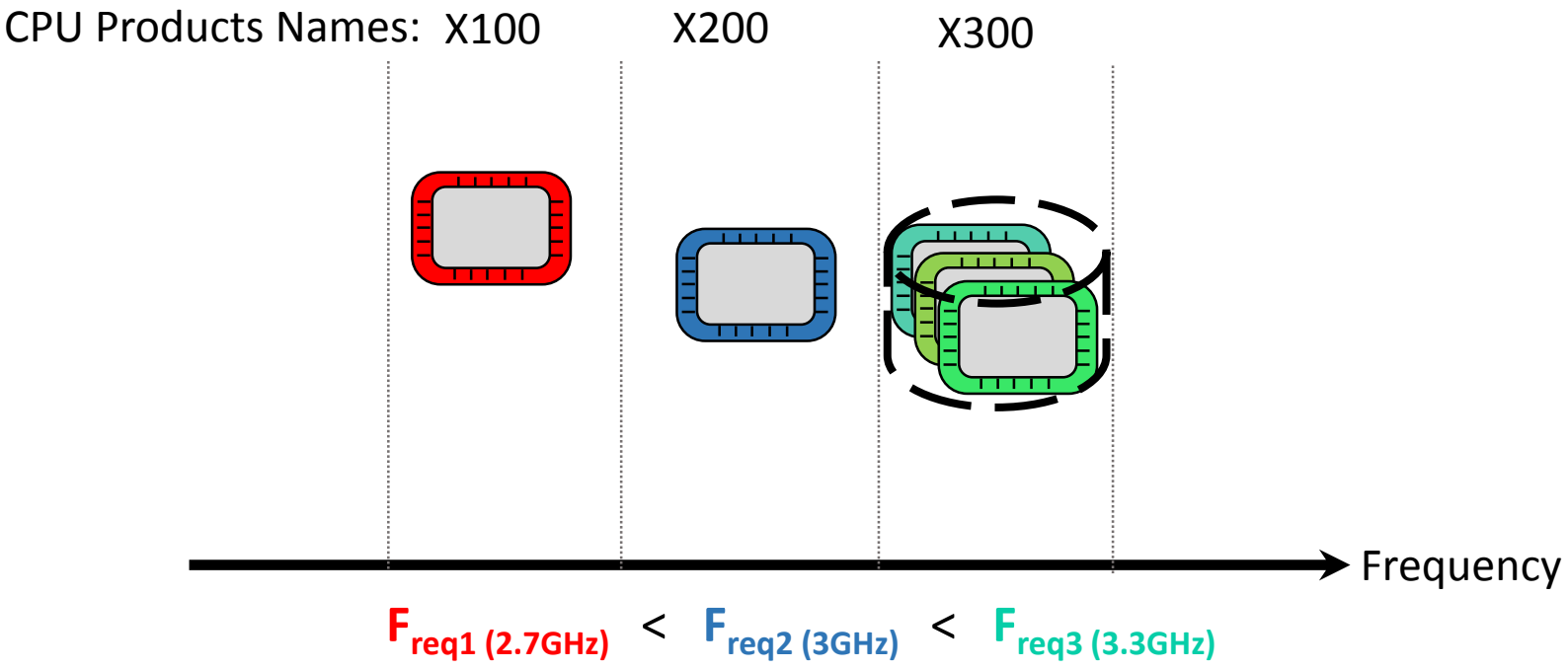
Based on [HEsm.ISCA11, ITRS13, JKoo.AHC11],

Limited Voltage Scaling Factors

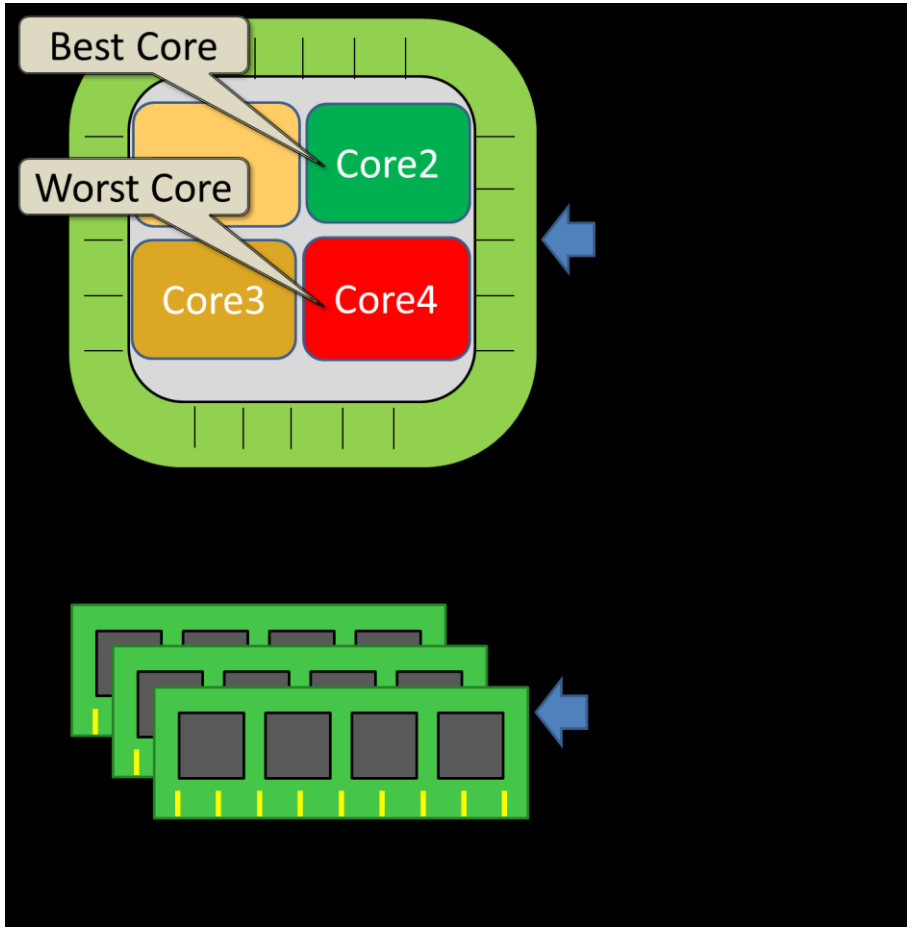
- Unpredictable issues at small technology nodes
  - Leakage
  - Vthreshold
- Voltage margins for
  - Process variations
  - Voltage Noise

Process Variations

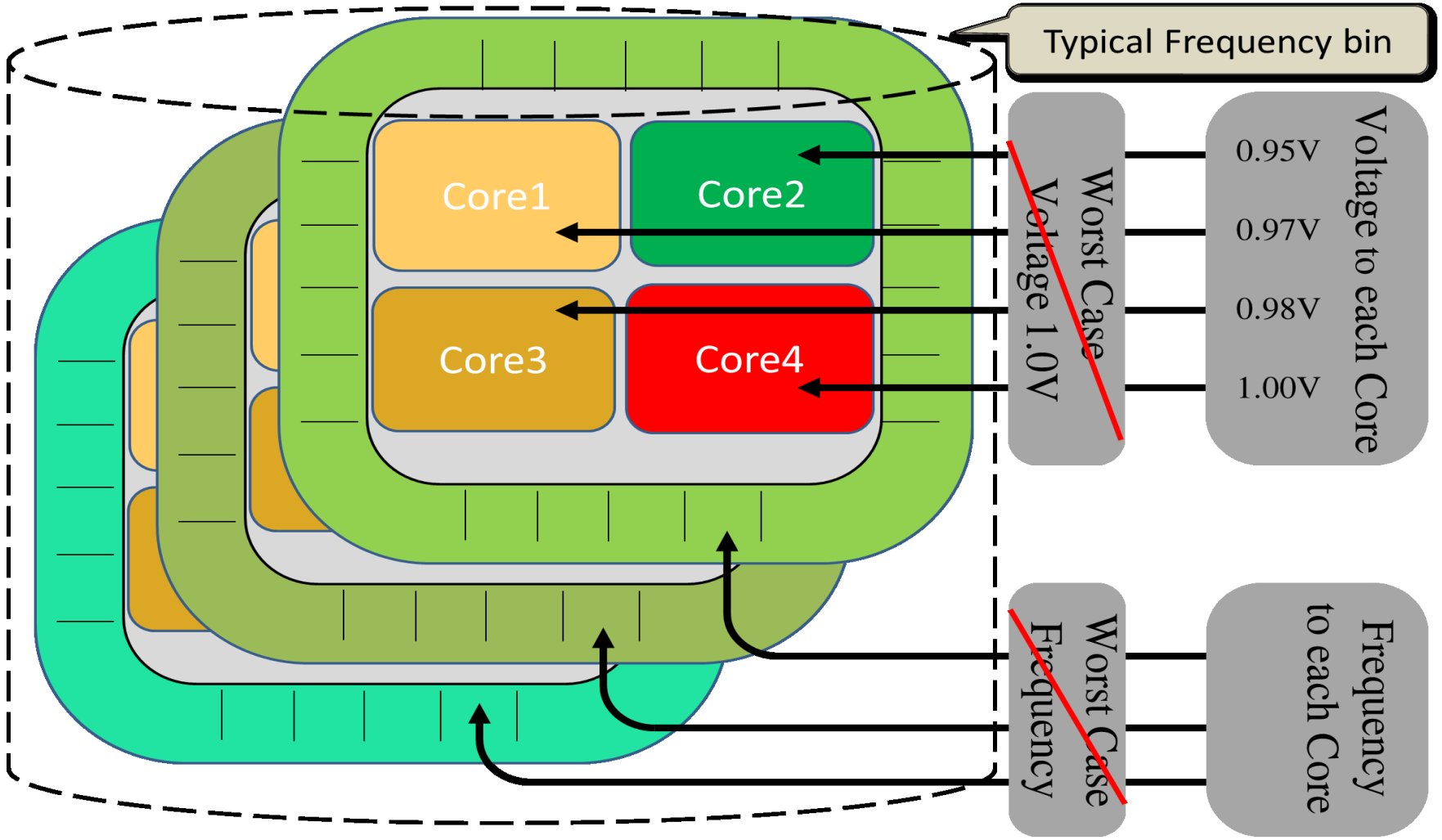
Across Chips



Intra-Chip



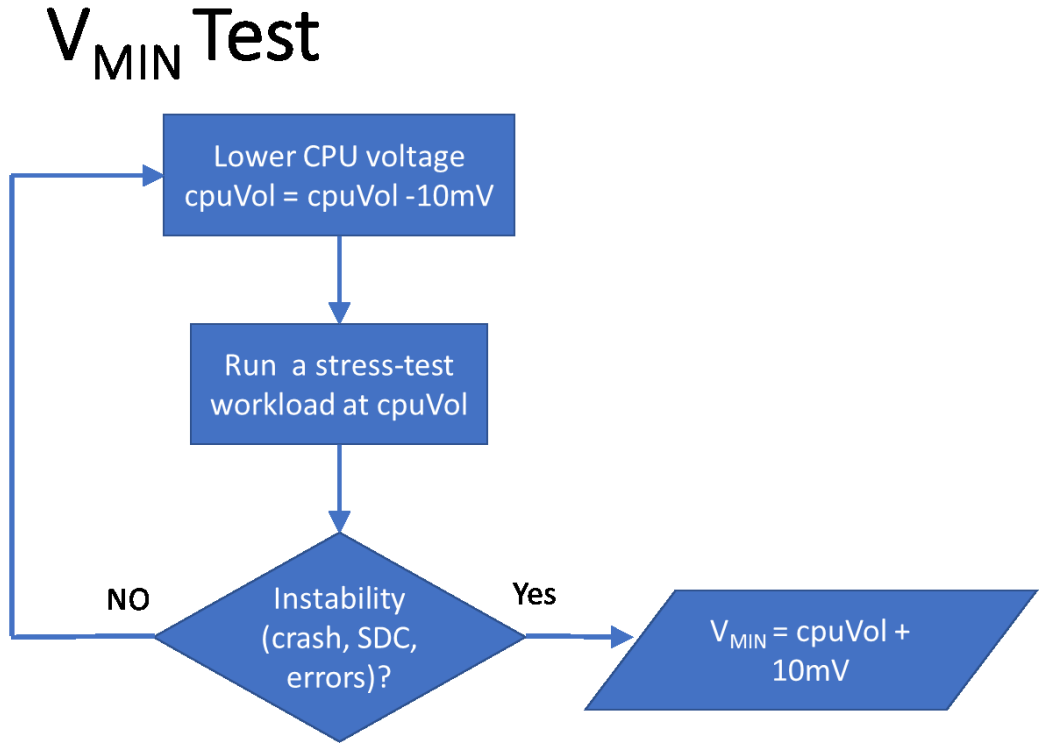
UniServer approach



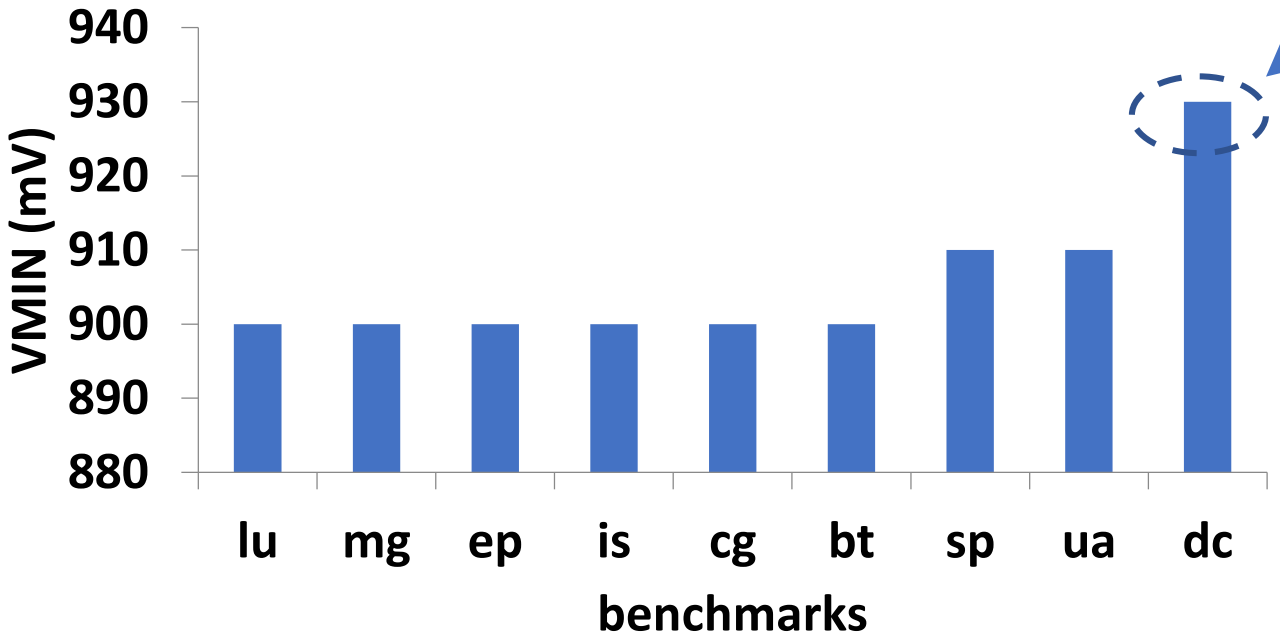
Depart from pessimistic operating points by revealing and exploiting at run-time the true capabilities of each CPU, DRAM, core,...

Dealing with hardware variations across chips/drams

- How to set the optimal voltage/refresh-rate for a given frequency for each chip/drams individually?
- Characterize each chip with V_{MIN} testing.
- Apply a different voltage to each chip based on the V_{MIN} test results



How to V_{MIN} test?
Test many benchmarks



- The highest Vmin is the chip's Vmin

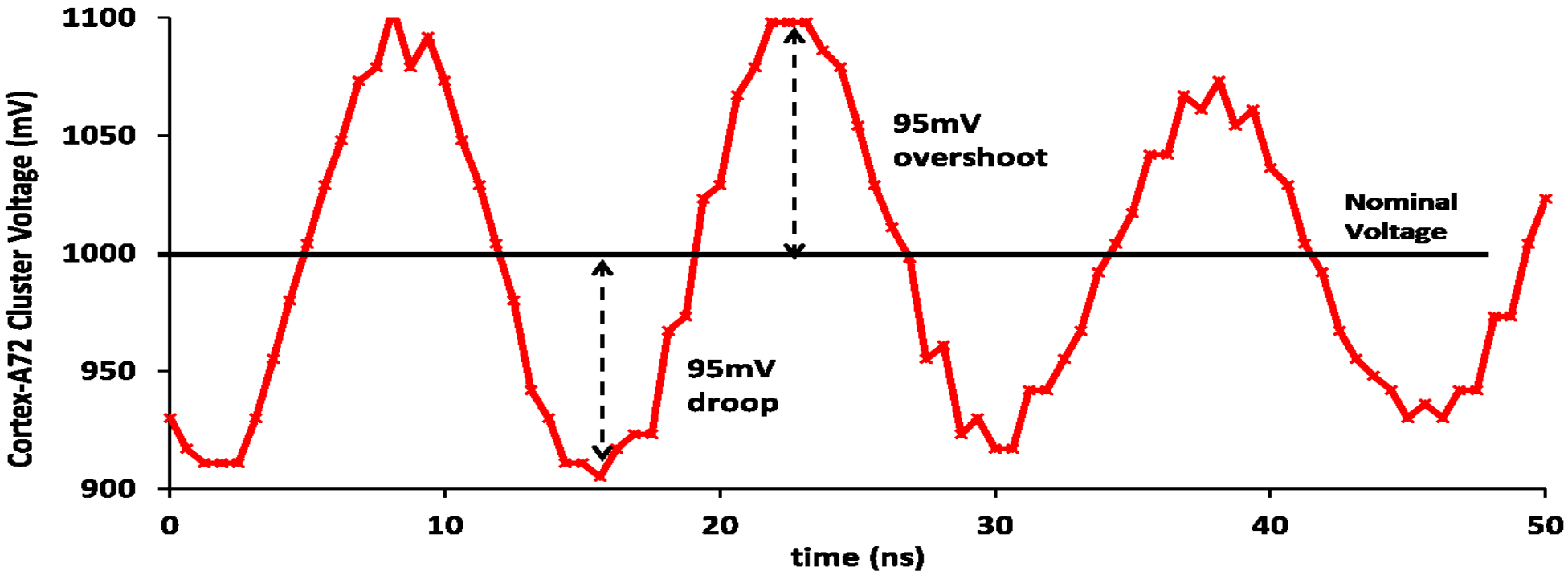
How to V_{MIN} test?
Test one worst-case benchmark/stress-test/virus

- Craft a virus that creates worst-case conditions and do a V_{MIN} run for the virus only
- We need CPU, LLC (last level cache) and DRAM viruses
- DRAM and LLC viruses:
  - Attempt to fill the SRAM/DRAM cells within patterns that maximize the probability of bit-flips
- CPU viruses:
  - Attempt to maximize **voltage-noise**. Note this is different from maximizing power e.g. Prime95

Limited Voltage Scaling Factors

- Unpredictable issues at small technology nodes
  - Leakage
  - Vthreshold
- Voltage margins for
  - Process variations
  - Voltage Noise

Voltage-Noise in CPUs



Caused by sudden variations in CPU power consumption

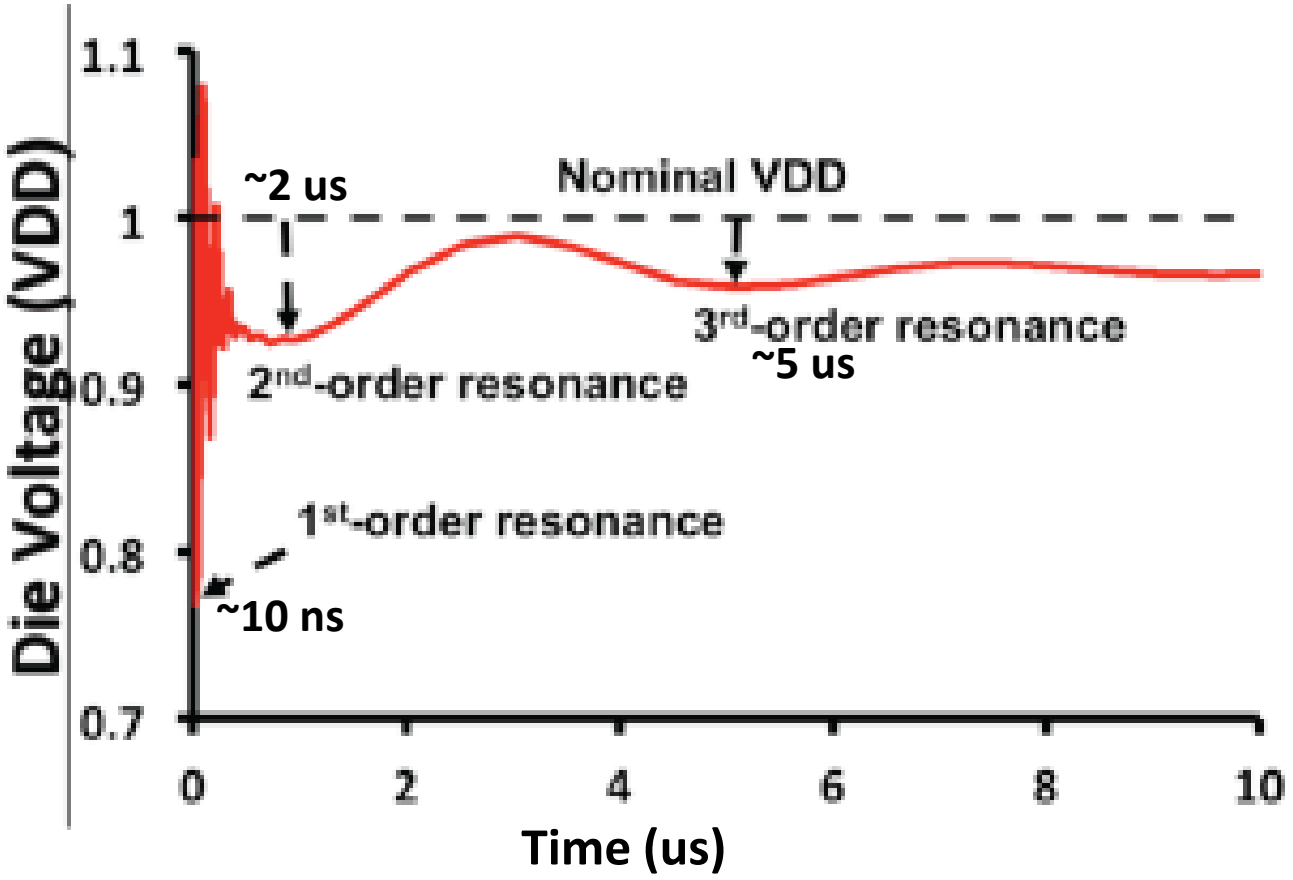
Voltage-noise sources:

Low-power techniques e.g. clock-gating and power-gating

Pipeline flushes (e.g. due to branch miss prediction) followed by high power activity cause

Activity switching between **high** and **lower power** at a rate equal to the **PDN 1st resonance frequency**

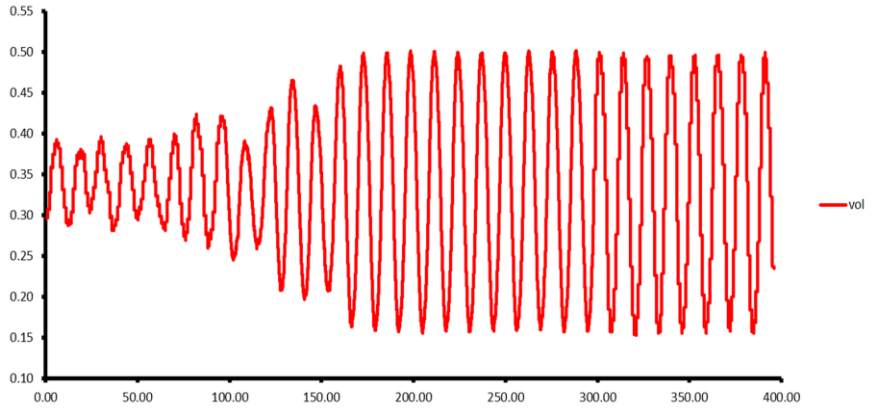
Power-Delivery-Networks 101



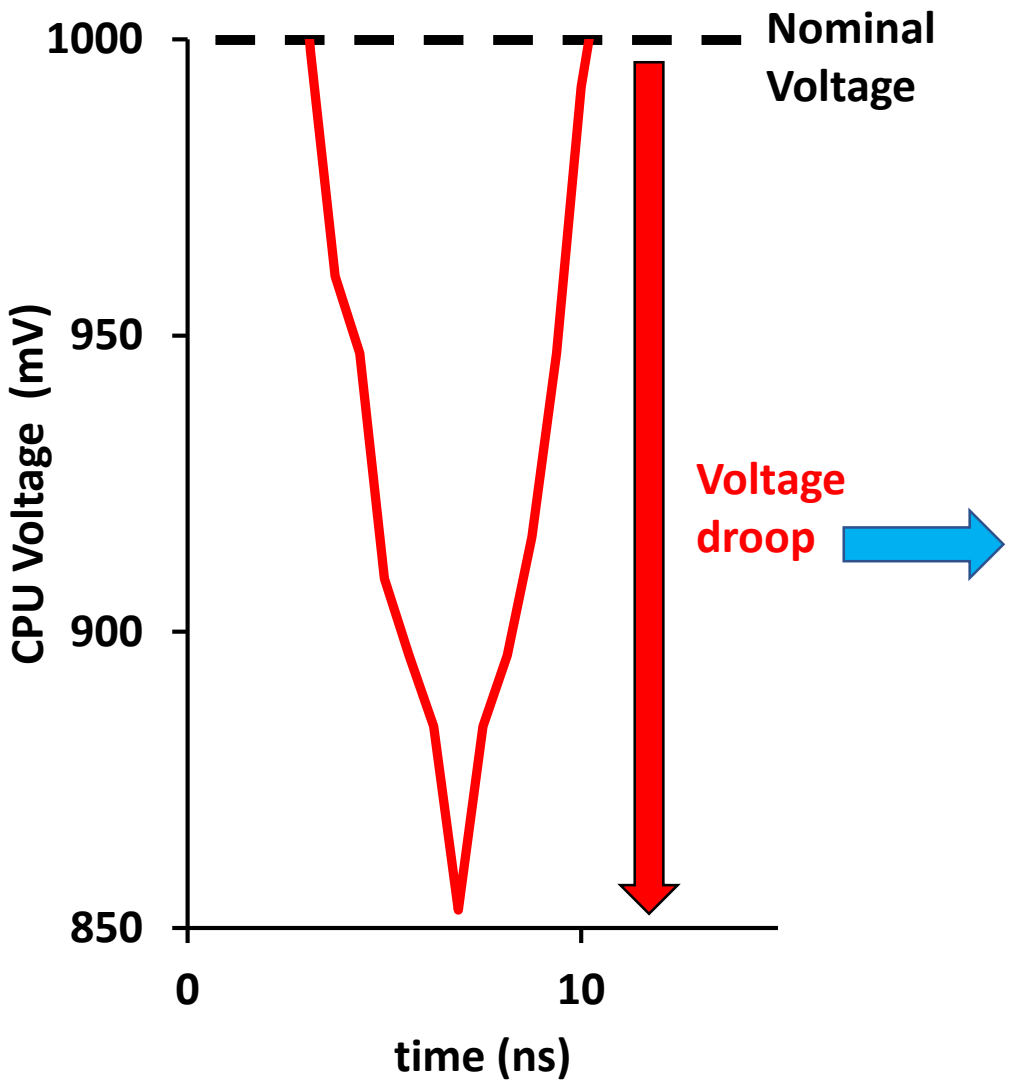
PDN reacts to large current stimulus with a response that has 3 dominant frequencies (resonance frequencies)

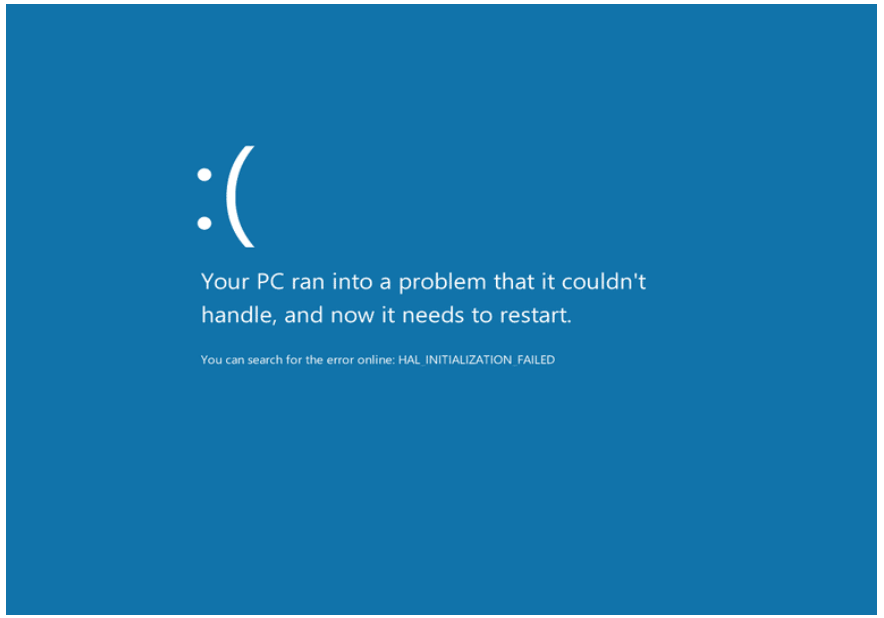
The largest droop (1st order resonance droop) occurs at ~10 ns

Repeated large current (I) draw at 10ns amplify voltage-noise. **This is what voltage-noise (dI/dt) viruses attempt to do.**

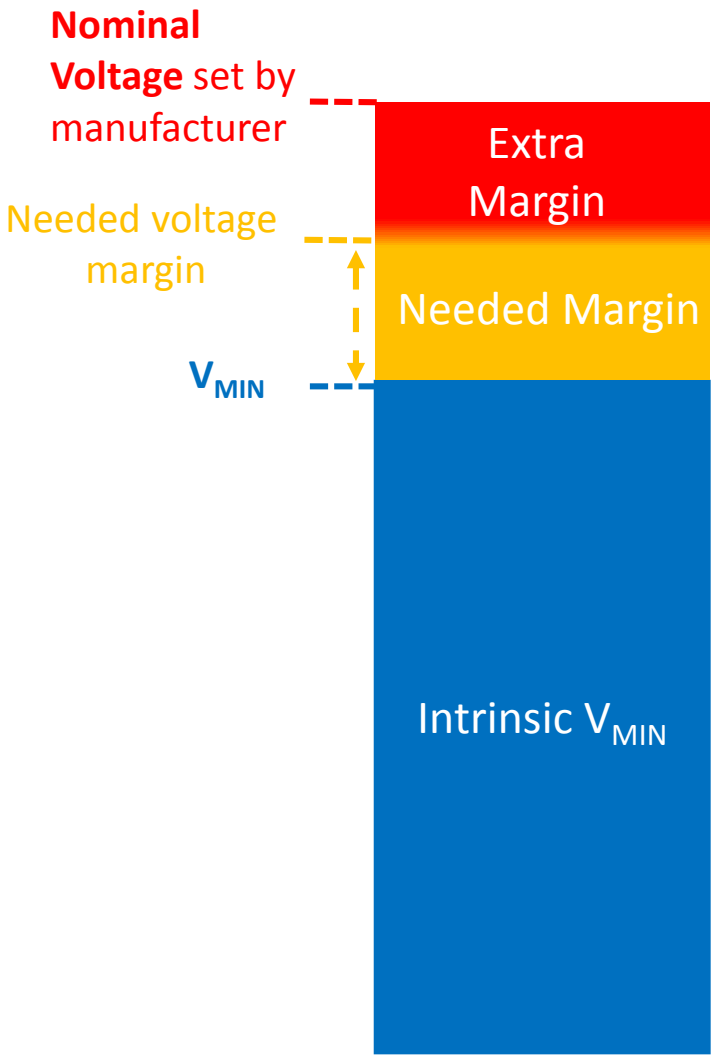


Why voltage-noise is bad?

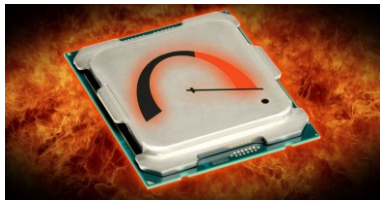




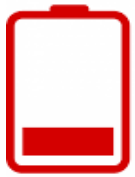
How to deal with voltage-noise?



Excessive Margining



Heat issues



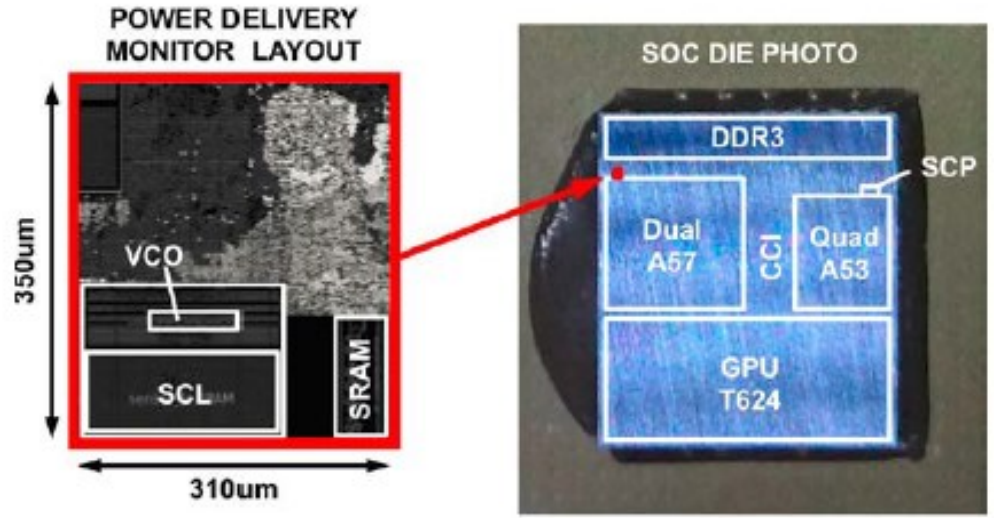
Energy inefficiency

How to shave the unnecessary margin?
Use **dI/dt stress-tests**, they are good **stability tests** for **V_{MIN} testing** because they drop the voltage very low
The V_{MIN} of the virus can dictate the nominal voltage

Voltage-noise (dI/dt) viruses development

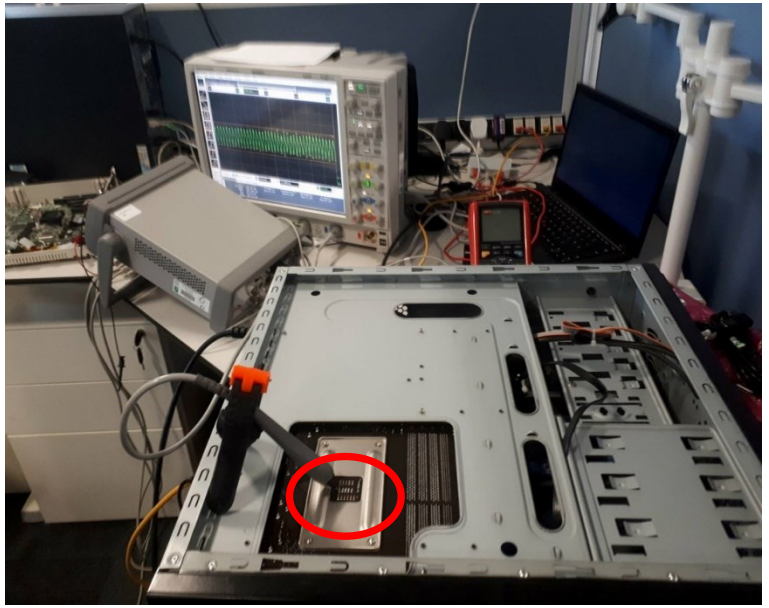
High-bandwidth voltage-measurements

On-Chip Circuits



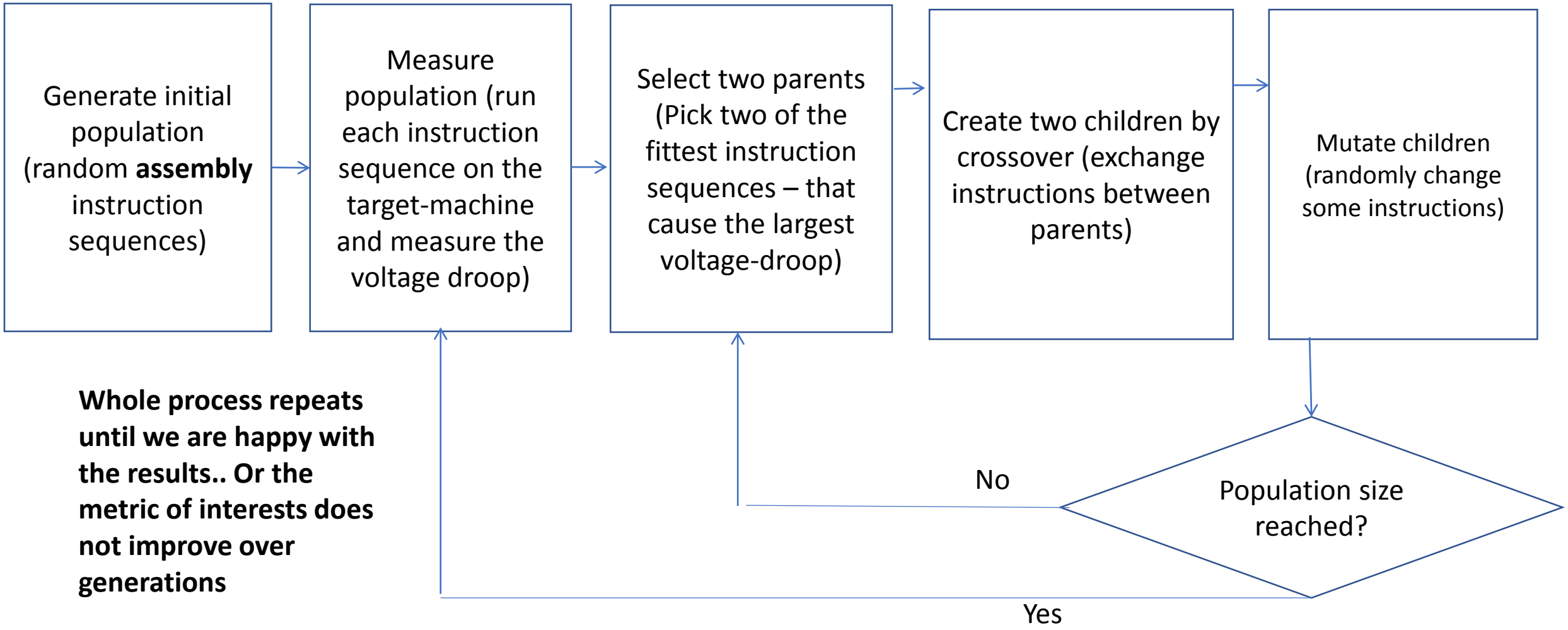
SOURCE [ARM ISSCC 2015]

On-Package Measurement Points



- To develop dI/dt viruses **high-bandwidth voltage-measurements** are required to measure the virus effectiveness and progress. Otherwise have to rely on V_{MIN} (which is very time consuming)
- **Genetic algorithms (GA)** to find the type and order of instructions that maximize voltage-noise. Manual virus crafting is possible but time-consuming

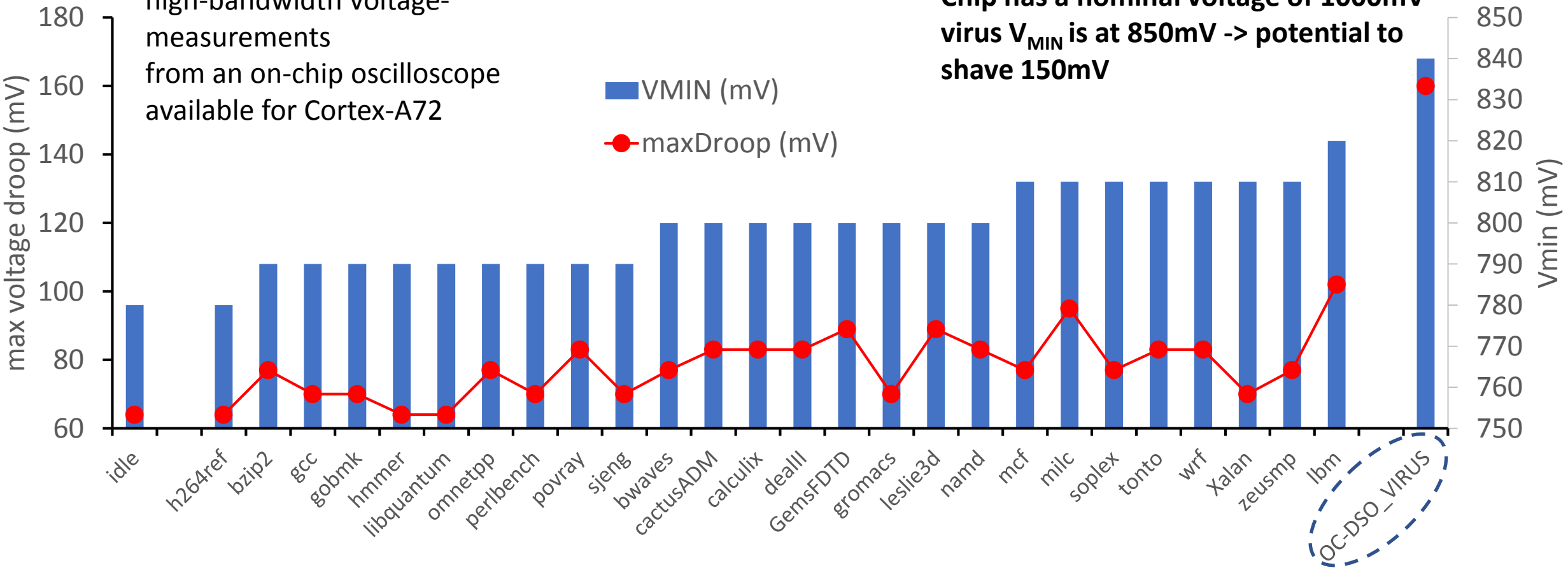
Genetic-algorithm for dI/dt virus generation



GA dI/dt virus vs SPEC benchmarks on Cortex-A72

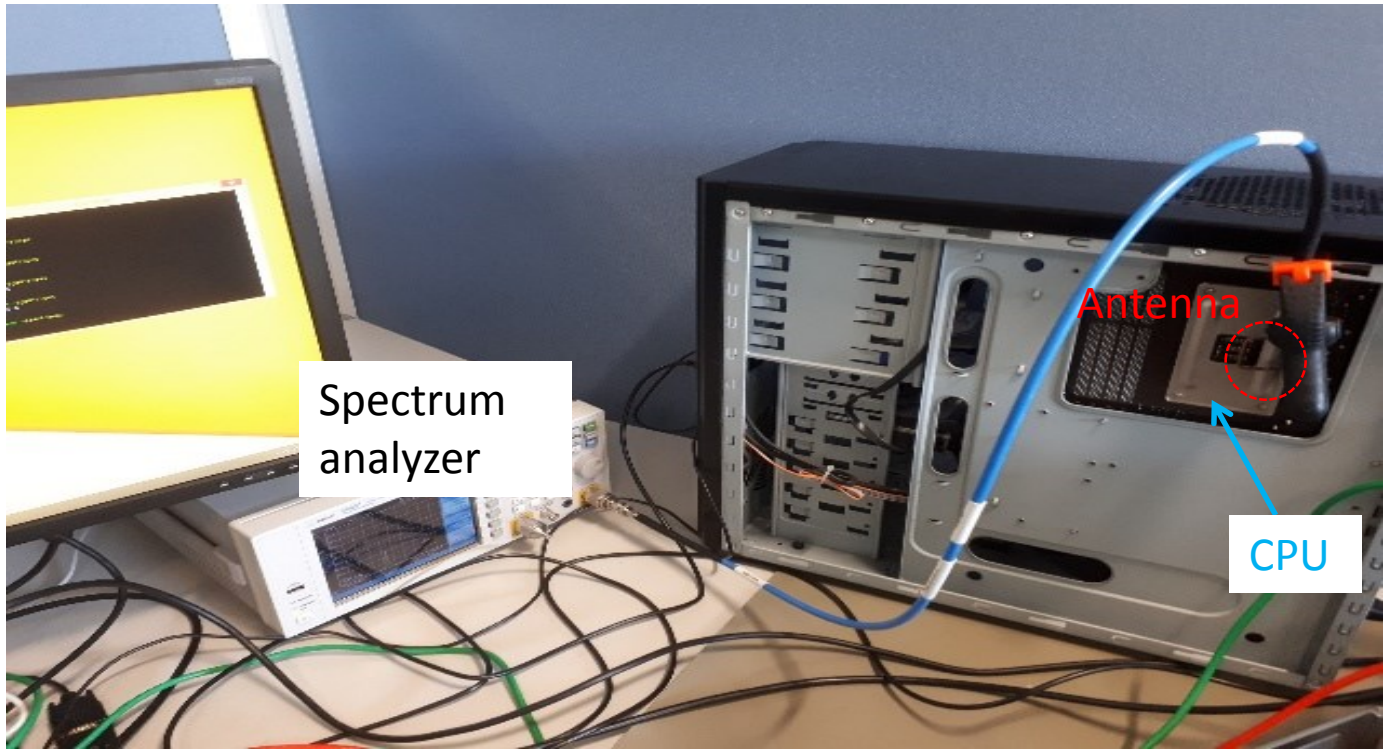
GA optimization was driven by high-bandwidth voltage-measurements from an on-chip oscilloscope available for Cortex-A72

Chip has a nominal voltage of 1000mV virus V_{MIN} is at 850mV -> potential to shave 150mV



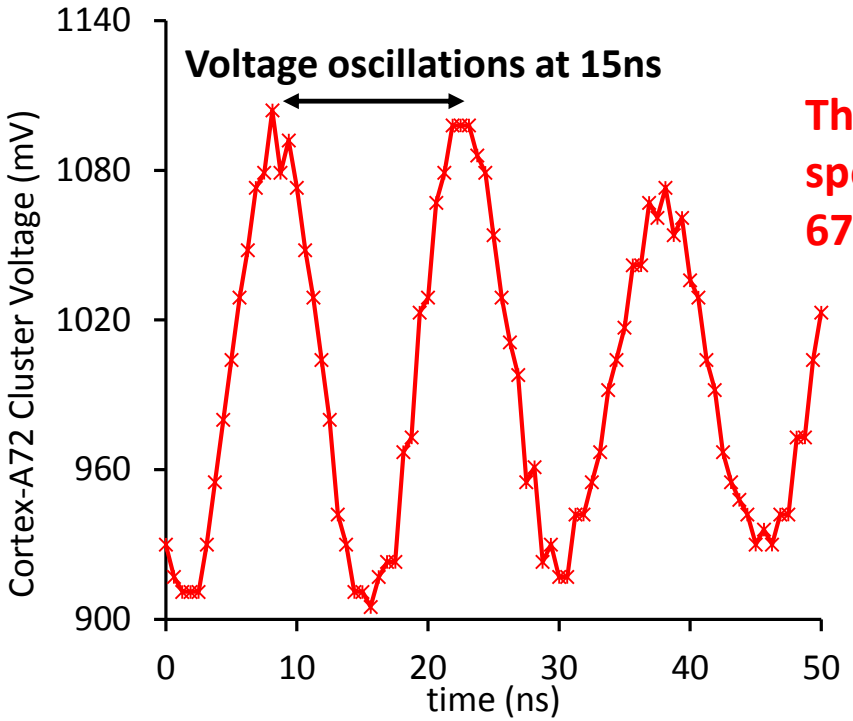
GA virus has higher V_{MIN} and causes higher voltage droop

Novel methodology for high-bandwidth voltage-noise measurements

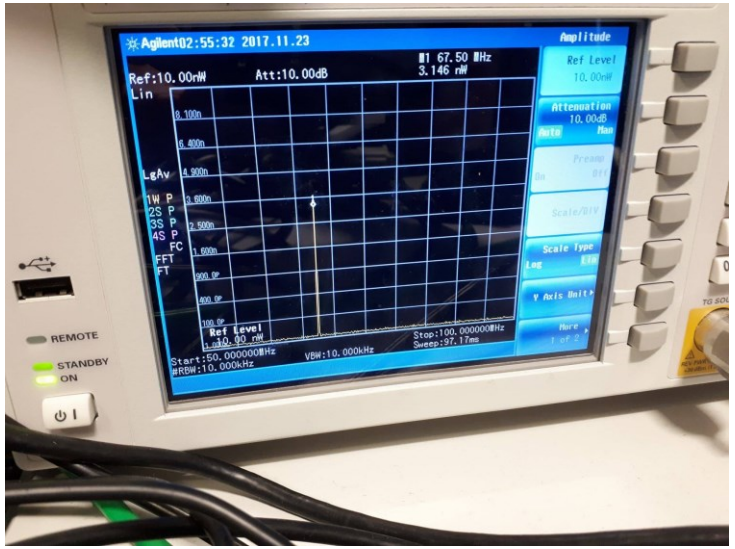


- X-Gene2 validation board doesn't support high-bandwidth voltage-measurements, had to find an alternative approach for generating dI/dt viruses

Voltage-noise oscillations manifest as EM spikes

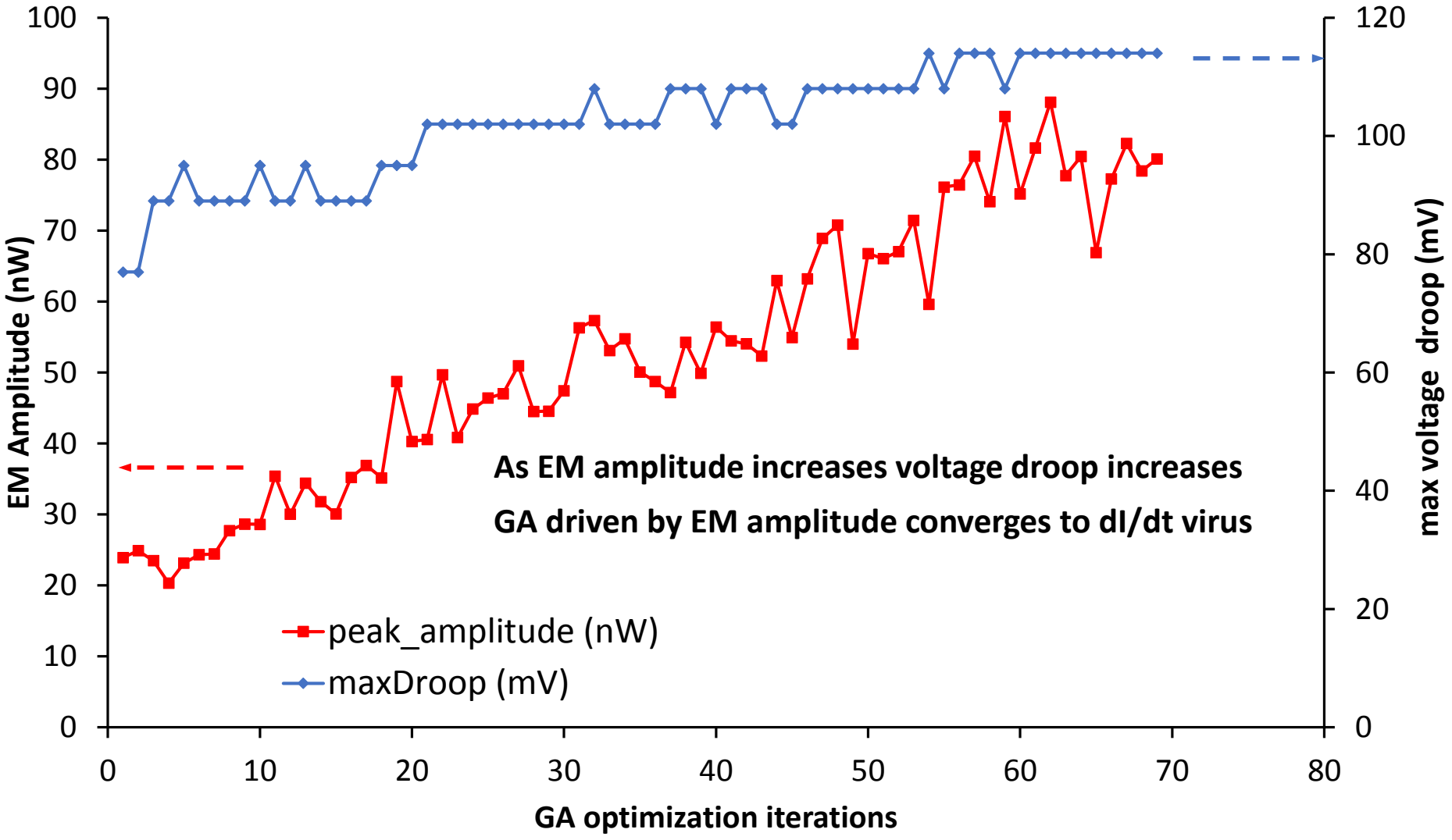


This manifests as spectrum spike at 67MHz (1/15ns)

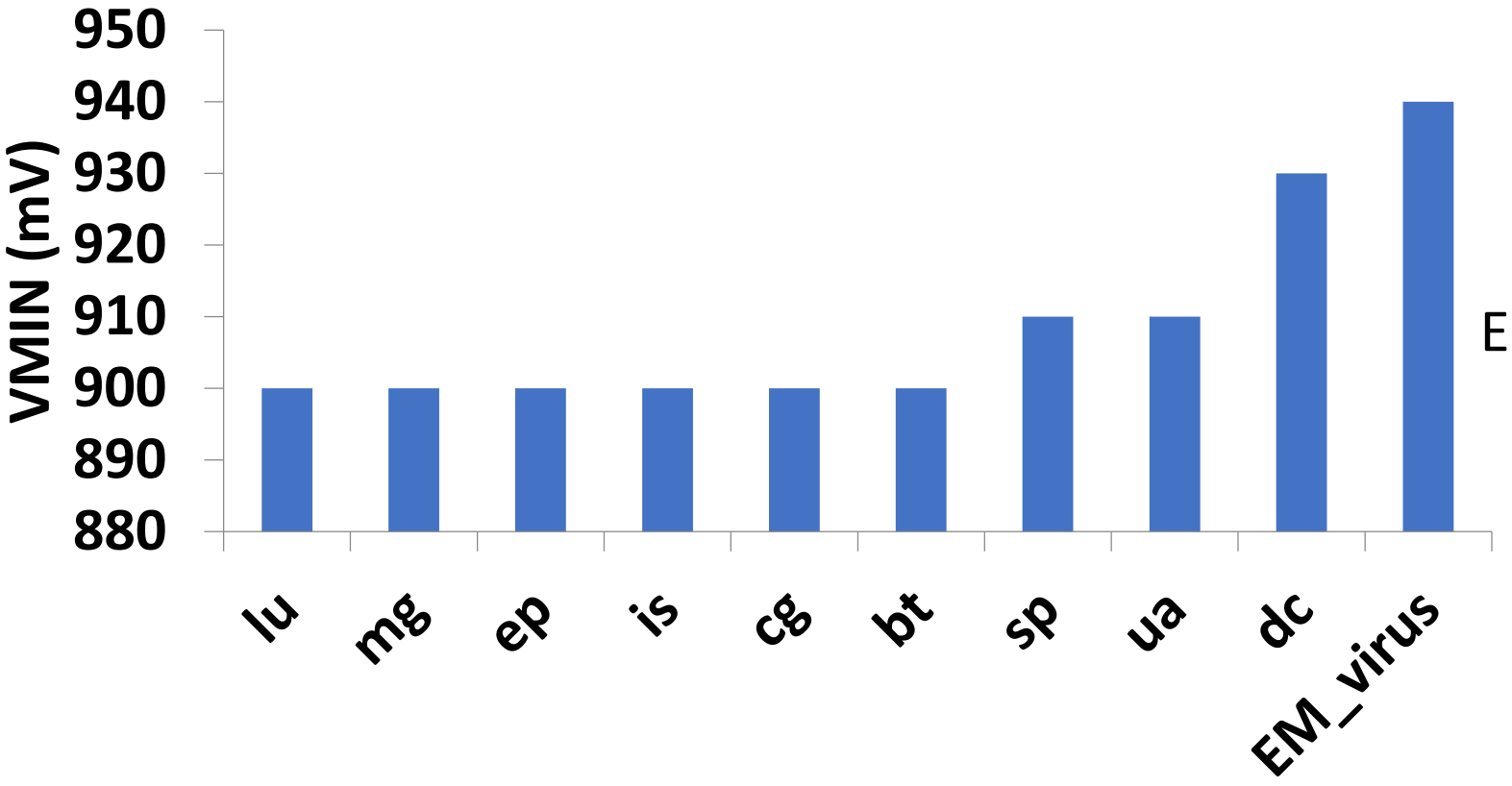


Higher-Amplitude EM signals => Higher voltage noise

GA driven by EM

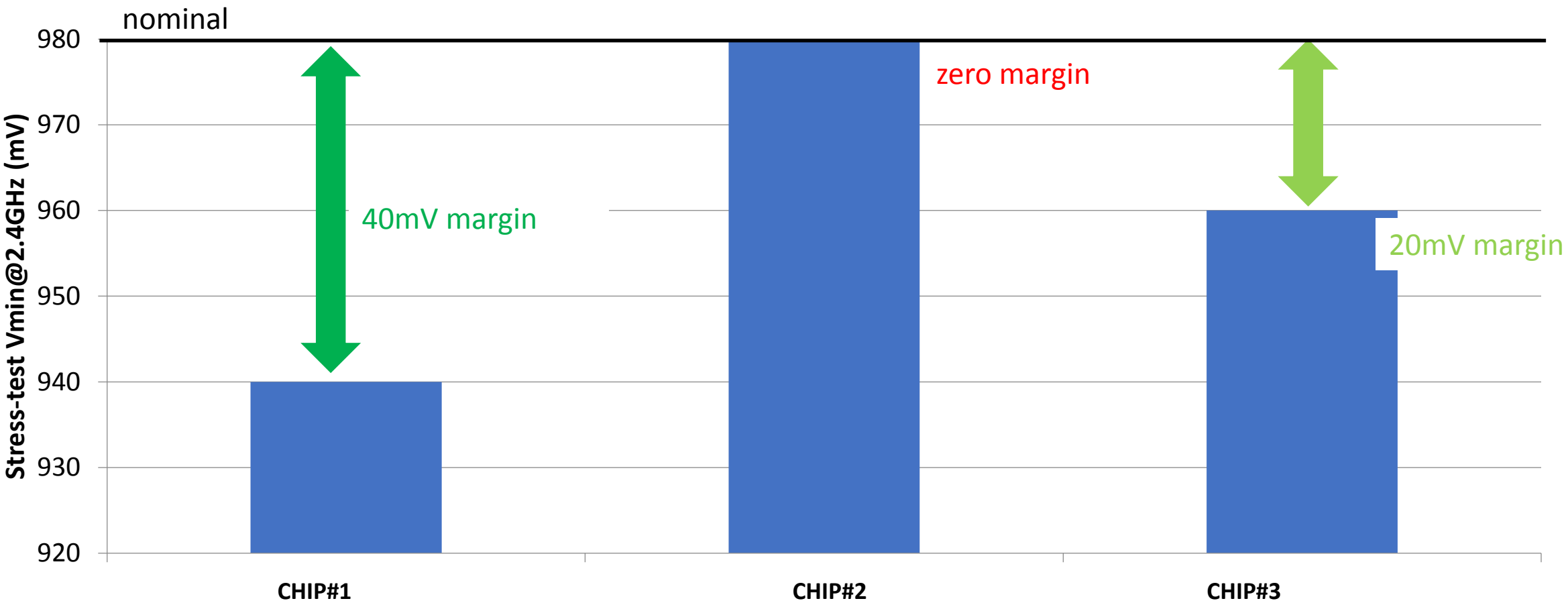


EM virus on X-Gene2 V_{MIN} vs NAS benchmarks



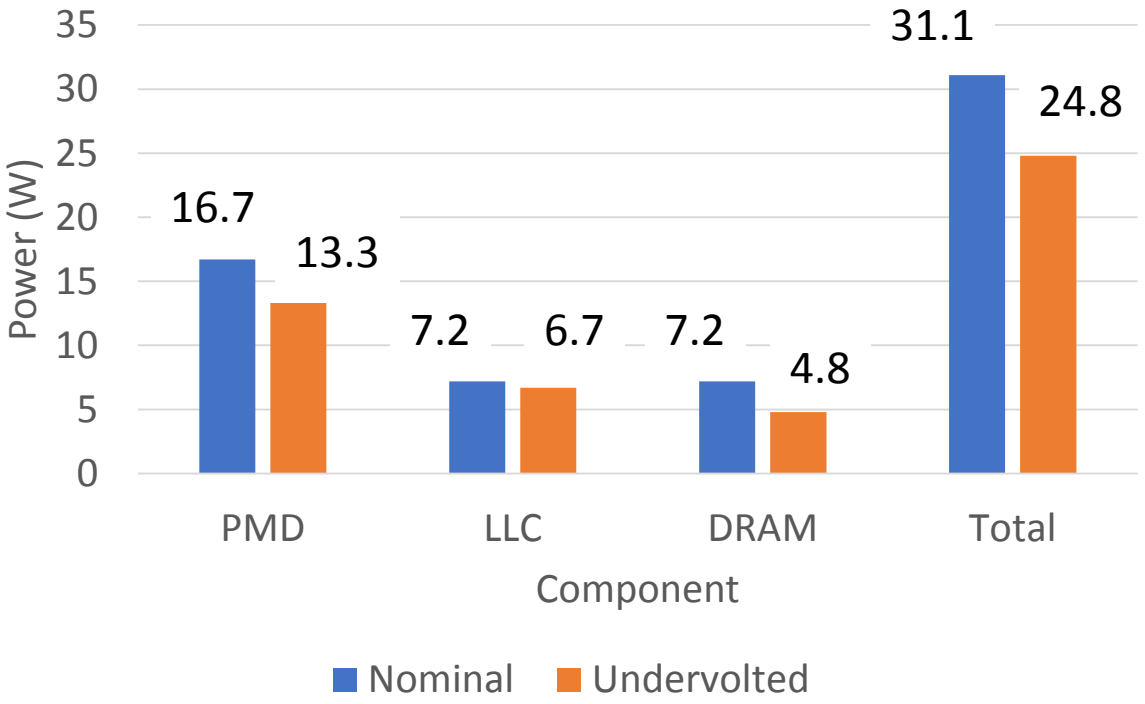
EM virus has the highest V_{MIN}

Exposing hardware heterogeneity with EM virus



We can lower the voltage on chips #1 and #2 to save power

Overall savings by undervolting all components



PMD voltage reduced 50mV
LLC voltage 30mV
DRAM refresh rate relaxed 35X

20% overall power savings