



SCALING VERIFICATION FOR HIGH-PERFORMANCE CHIPS

Why Hardware-Assisted, Real-Time Visibility Isn't Optional

Frederic Leens - Exostiv Labs

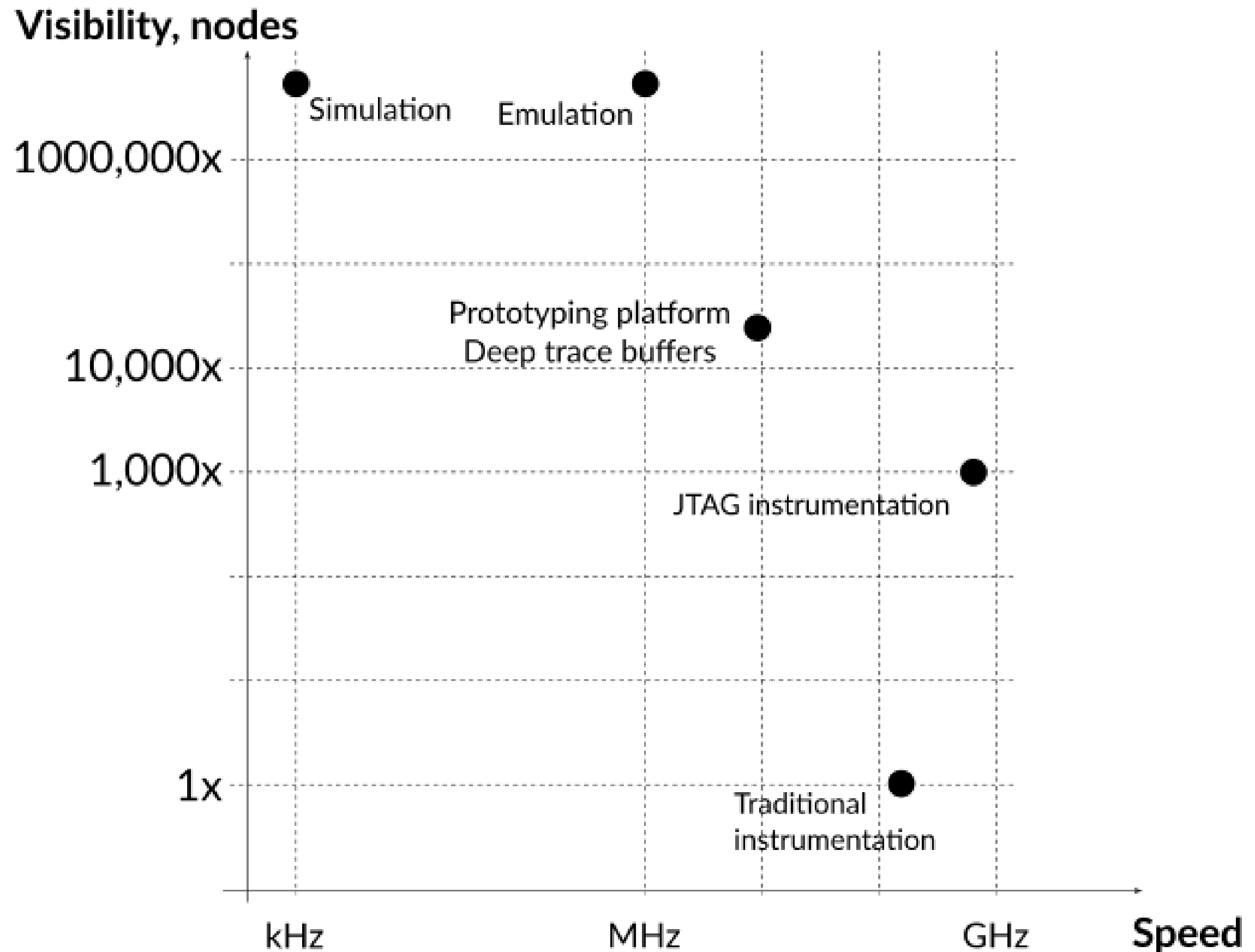
The background of the slide is a blue-tinted photograph of a complex electronic circuit board. A magnifying glass is positioned over a central integrated circuit (chip), highlighting its intricate details. The overall aesthetic is technical and focused on hardware verification.

Hardware-assisted verification cases with the need of:

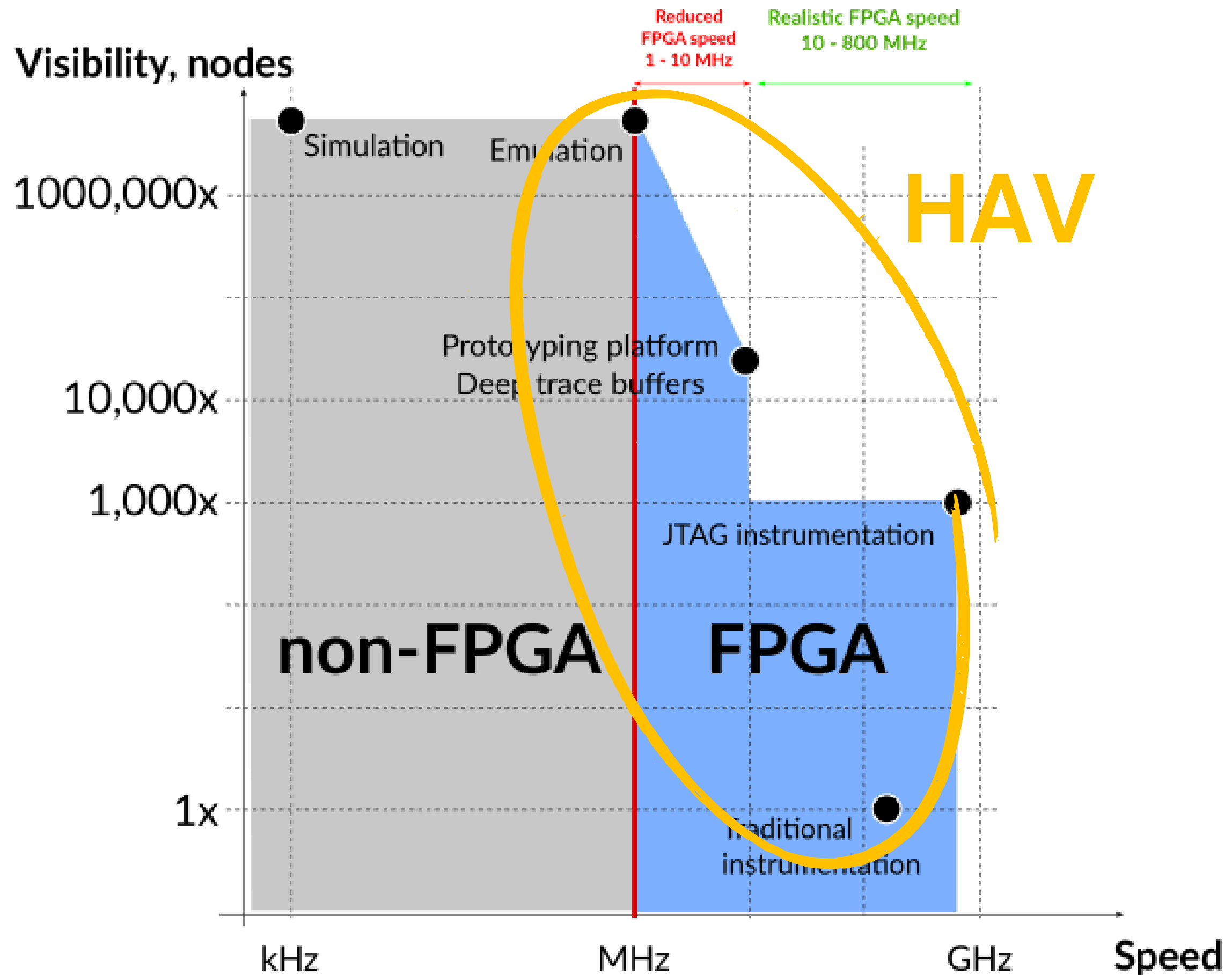
- + **Massive visibility**

- + **At system speed**

How tools compare



How tools compare



HAV visibility at speed is traditionally hard

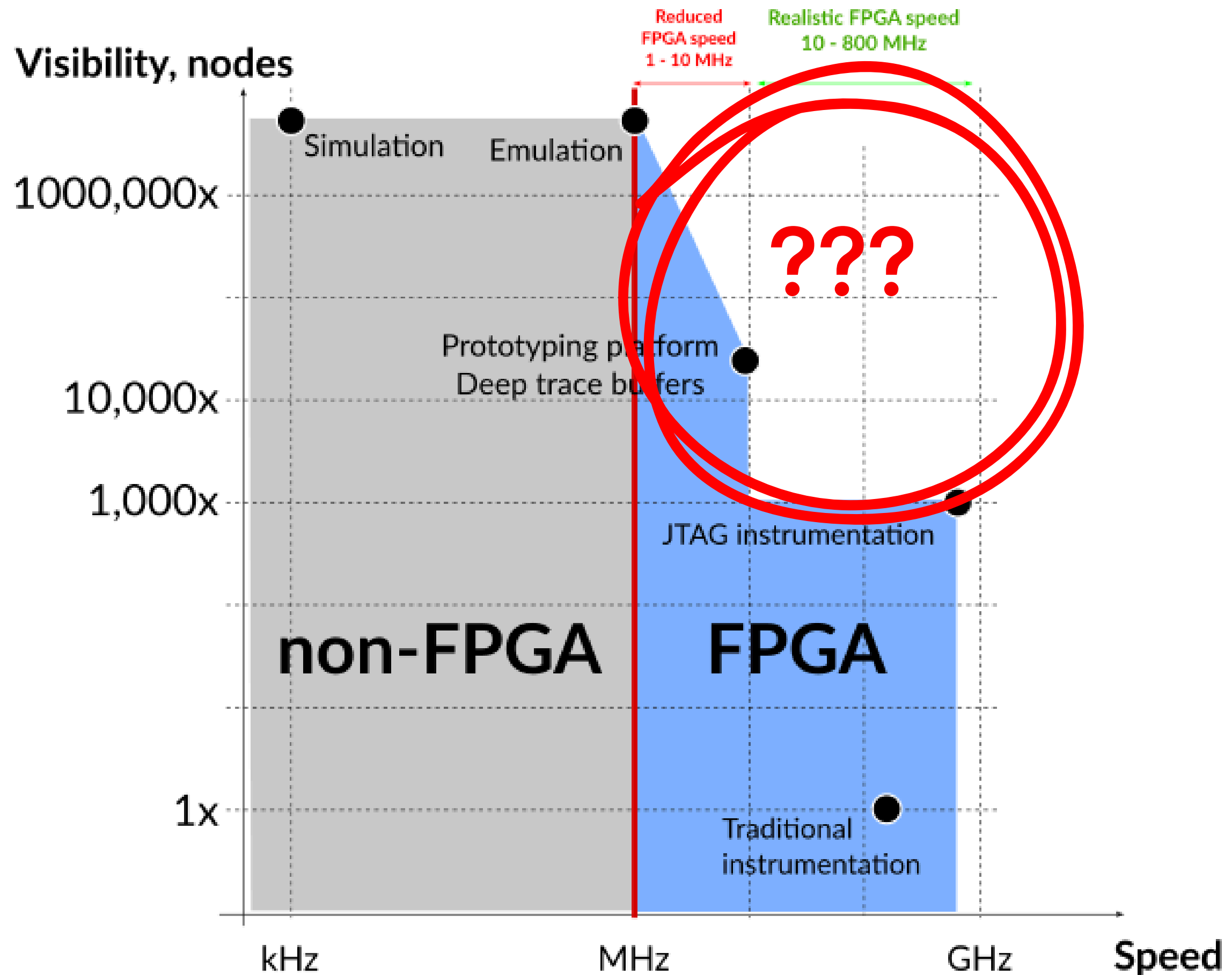
Emulation :

- **No at speed capability.**
- 40+B gates
- Typ. 1 MHz (claims up to 10 MHz)
- 'Full' visibility
- Weak on interface (hence realism).
- Requires hard work on modelling
- Can be 'clogged' with small jobs
- (Only?) solution for very large systems.

Prototyping :

- **At speed capability if hardware is leveraged for prototyping and tools.**
- 15+B gates
- Typ. 2 MHz up to 50 MHz
- 500+ MHz on single FPGA
- Realism potentially very high.
- Partitioning is key to performance.
- FPGA expertise is key.
- Solution for small to large systems.
Can be designed in house.

Is this reachable?
Is it necessary?



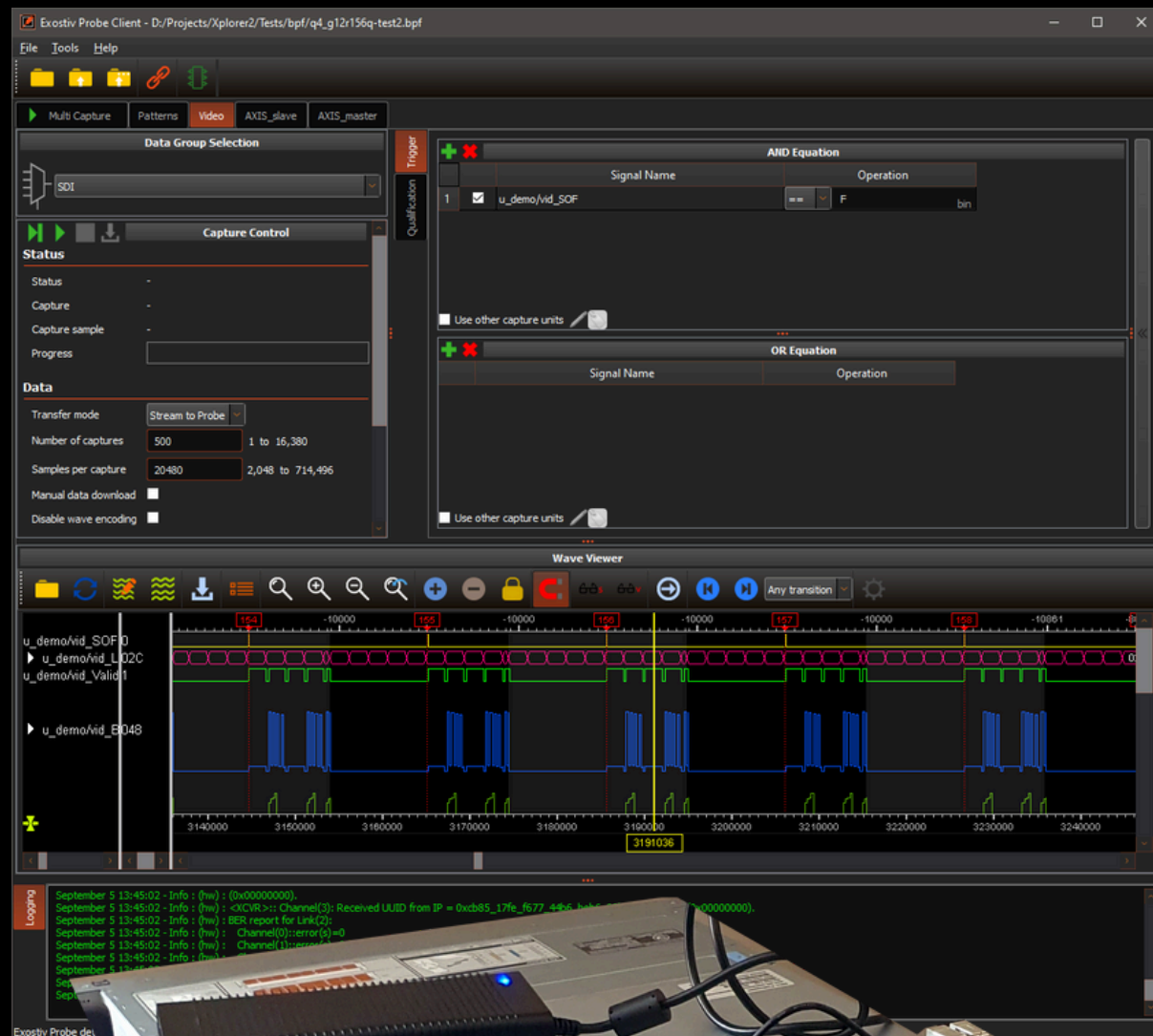
Quotes from the field... **Is it necessary?**

- ‘1 MHz to 500 MHz is just a multiplication factor...’
- ‘We look at system / model / transaction instead.’
- Software can run on emulators now.
- Use models. It’s mathematic.



We make standard products for productivity gains

- Since 2016
- All our clients use FPGA prototyping or emulation
- ASIC - SoC - IP - FPGA OEMs
- Tools are integrated in their flow



Case 1 : SoC with communication interface

Complex SoC with external world interface (USB)

- Need to check new protocol IP operation & interoperability with legacy existing equipment.

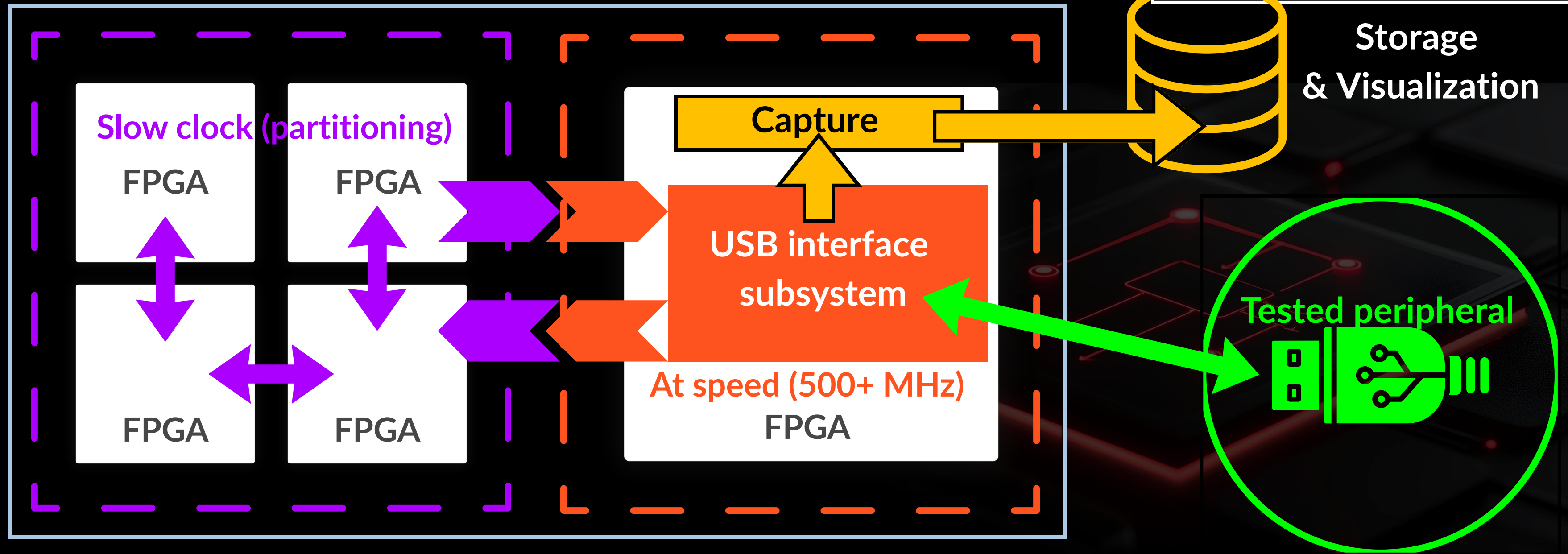
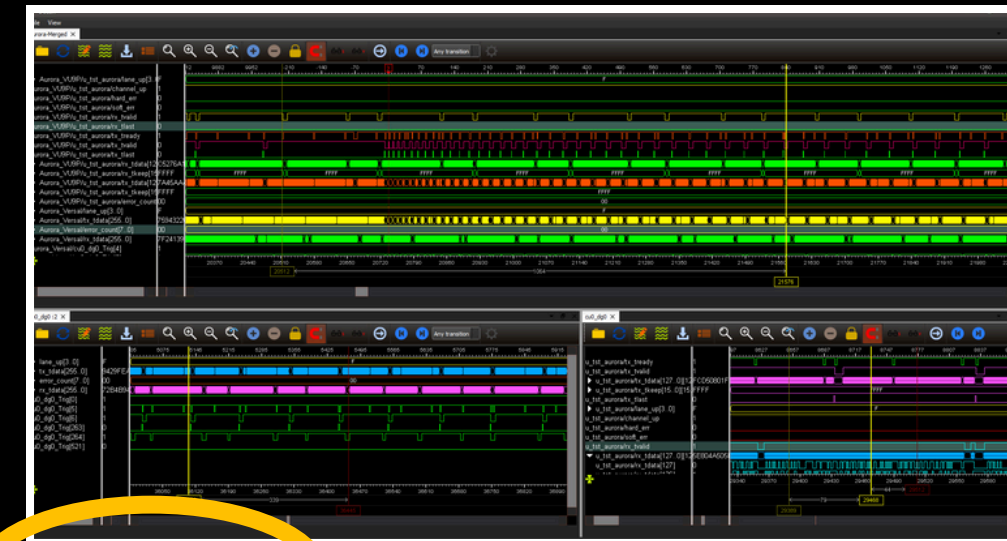
Challenges:

- Need to validate realistic software execution times
- External equipment cannot be slowed down
- Pure model impracticable: goal is checking real world interop.
- **Emulation is too slow for software and forces slowing down the environment, which cannot be achieved or costs realism.**



Case 1: SoC with communication interface

- Need for realistic nr of cycles : **external traces storage (GB)**
- Directed / manual FPGA partitioning, fitting interface subsystem in a single FPGA : **minimal instrumentation footprint**
- At speed operation : **high speed sampling (500 MHz+)**



Case 2: Specifying complex AXI structure for SoC

Next generation VR headset design

- GTM planning driving need to decide about technology choices
- Re-using existing HW platform vs. Design new HW platform?

Key AXI IPs prototyped in AMD evaluation kits (Virtex Ultrascale+)

- Complex AXI bus structure
- Clock frequency ++
- Embedded software execution

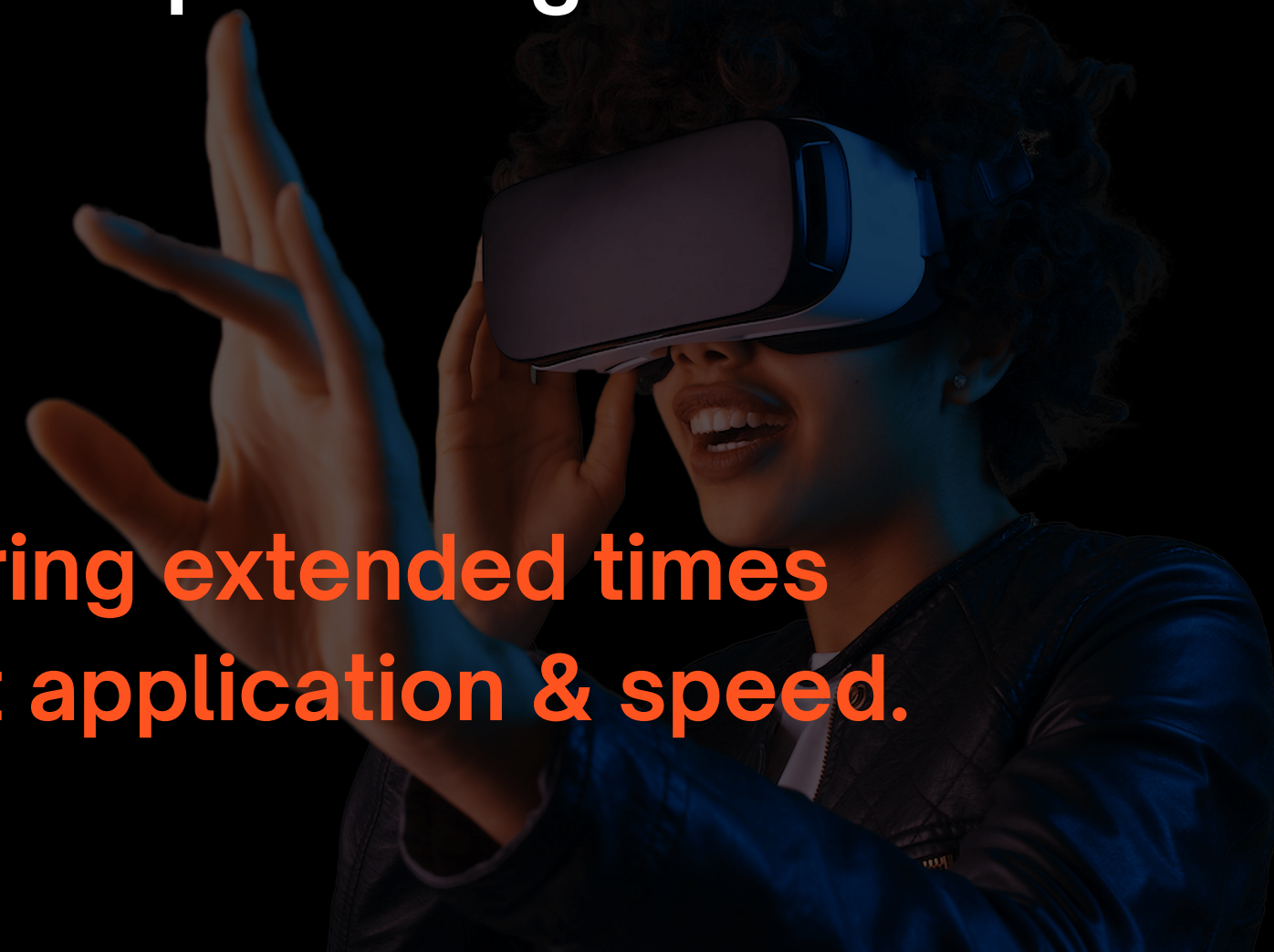


Case 2: Specifying complex AXI structure for SoC

Demonstration that the existing set of IPs and system bus structure can sustain higher execution rates.

- Decision to use the existing platform for the new product generation
- Ability to start mechanical design
- HW platform dev costs savings
- Faster go to market

The ability to observe the AXI bus structure during extended times allowed demonstrating its fitness for the target application & speed.

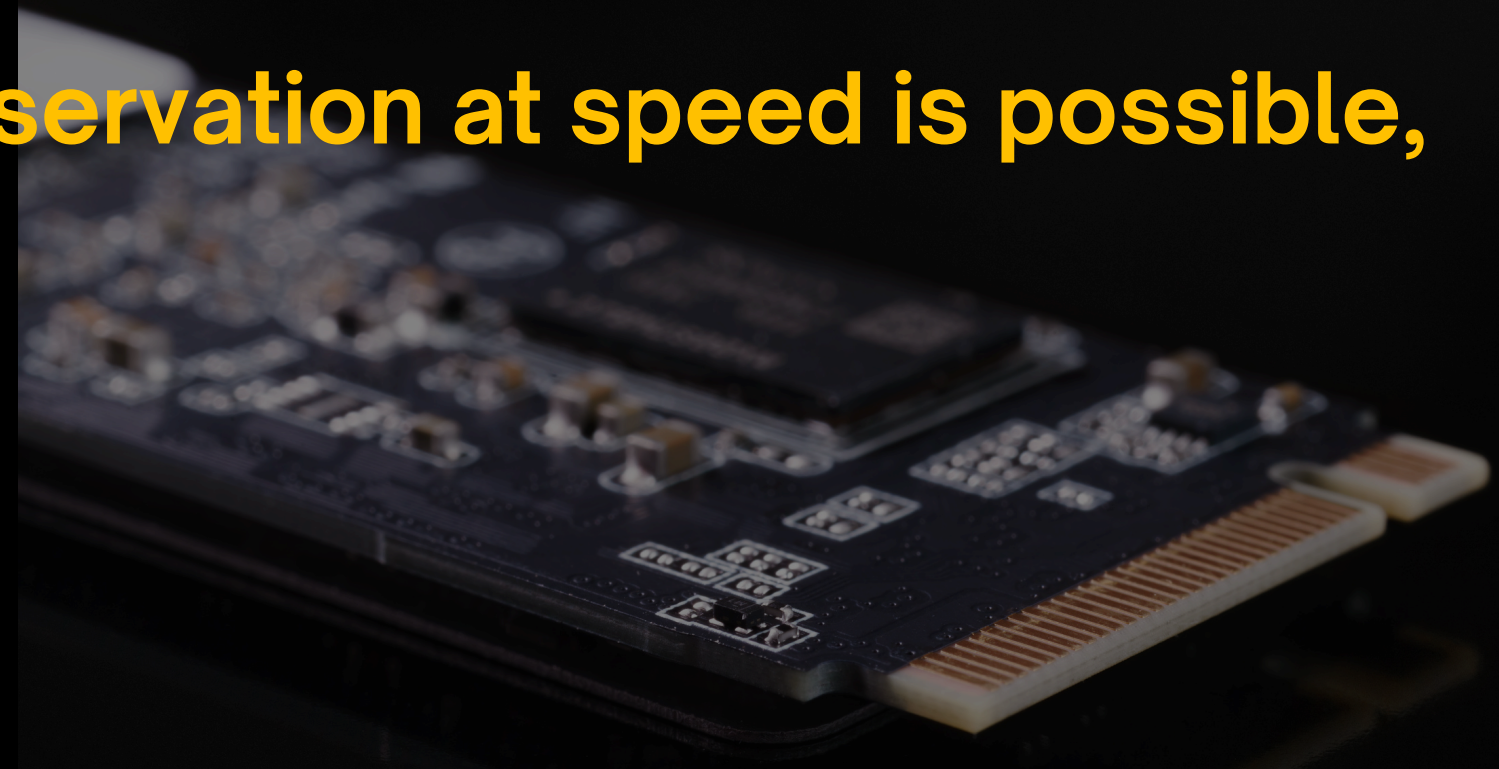


Case 3 : SSD controller

- Challenges before committing to ASIC :
 - Early firmware development
 - Performance evaluation
- Testing at or near speed before silicon is produced

Approach: implement the host and flash channel controller in FPGA.

The efficiency of the approach supposes observation at speed is possible, including on custom FPGA prototype.



Case 4: Complexity of RF source signal

Complex DSP algorithm development in FPGA

- Radio front-end (RFSoc device)
- Complex digital processing in the FPGA

Inability to model the complex signal > Simulation are insufficient.

- Signal capture in the digital world
- Recording of extensive databases with real-world analog signals
- Ability to tune the digital processing through software programming and tests various parameters & DSP architectures



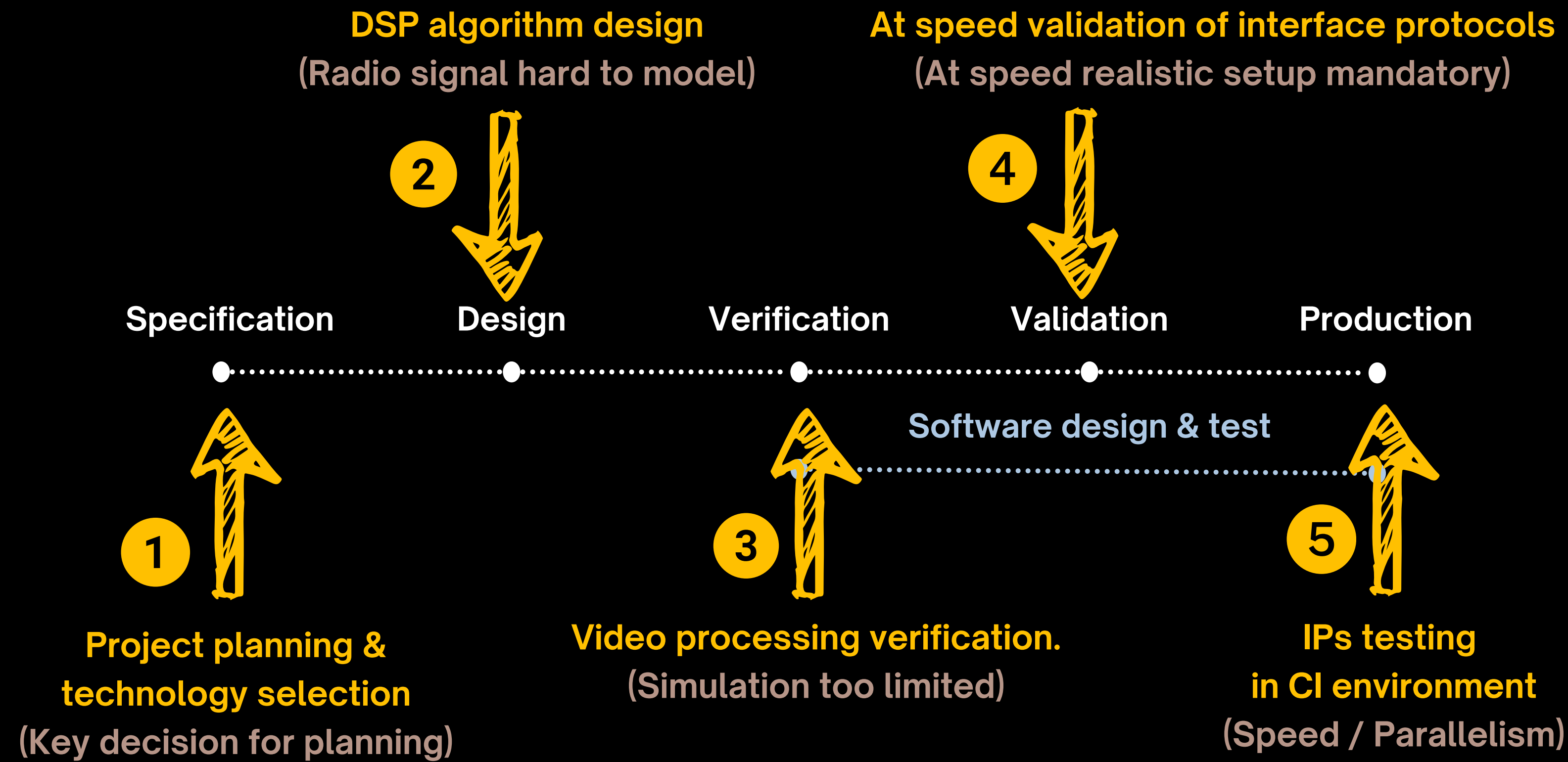
Case 4: Complexity of source RF signal

Definition of DSP parameters and architecture based on real-world data statistically more complete than theoretical models.

The ability to capture the signal in the digital world during extended times and reprogram the processing dynamically allowed design an empirical DSP algorithm ready to be deployed in a hardware platform.



Further on ... visibility at speed over the whole design flow



Other cases:

Contact me - frederic.leens@exostivlabs.com



Success stories

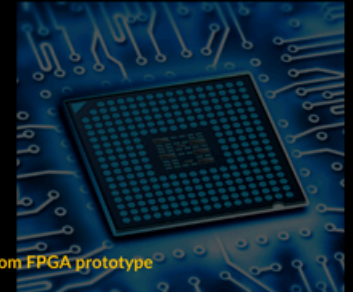
#1 - Large SoC Design

Case:

- Large SoC design with communication interfaces
- An interface IP is designed
- The communication protocol must be compatible with existing peripherals
- Simulation is not realistic: real world peripherals are not well modelled

Reasons for using Exostiv Blade:

- Only tool capable of capturing large data streams at speed of operation from FPGA prototype



2 - RISC processor IP

Case:

- A RISC processor IP is checked in hardware for proper software execution
- Need to capture software execution at bit level from the running IP and re-build software from waveforms

Reasons for using Exostiv Labs products:

- Work with proprietary FPGA board
- Capture massive data deep inside the FPGA prototype at speed of operation
- Automate captures (Python) and automate testing



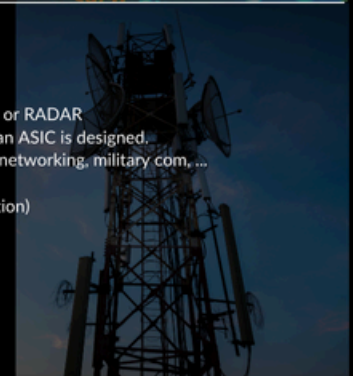
3 - Radio communications

Cases:

- Companies building products for radio communication, telecommunication or RADAR
- The technology is based on FPGA or FPGA is used for prototyping before an ASIC is designed.
- Communications from diverse sources and destinations, satellite, vehicles, networking, military com, ...
- Issue: modeling the radio (analog) signal is hard
- Processing happens in digital domain (after A/D conversion and demodulation)
- The resulting digital data is very large

Reasons for using Exostiv Labs products:

- Works with proprietary FPGA board
- Capture digital domain signal
- Capture digital samples at all taps of digital filters



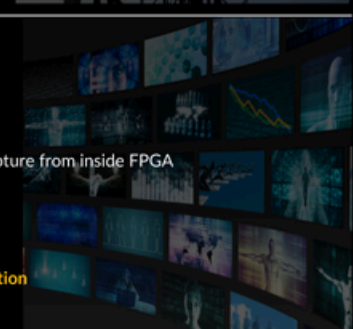
4 - Video processing

Cases:

- Digital video processing appliances in FPGA
- Data sets are too large for traditional tools, requiring extended history capture from inside FPGA
- Need to capture root causes after hours of testing

Reasons for using Exostiv Labs products:

- Work with proprietary FPGA board
- Capture massive data deep inside the FPGA prototype at speed of operation
- Automate captures (Python) and automate testing



www.exostivlabs.com

Key takeaways

- HAV visibility at speed is traditionally hard but not optional.
- Ideally: total visibility at operating speed (> 500 MHz).
- FPGAs can come close to real speed.
- Careful partitioning is essential for this type of scenario.
- At speed visibility requires significant hardware resources.

Hardware resources should allow you:

- Observe what you want (**REACH, bits**)
- Run at real speed (**SPEED, MHz**)
- Capture significant and useful data (**DEPTH, GB**)

Questions?

frederic.leens@exostivlabs.com



www.exostivlabs.com