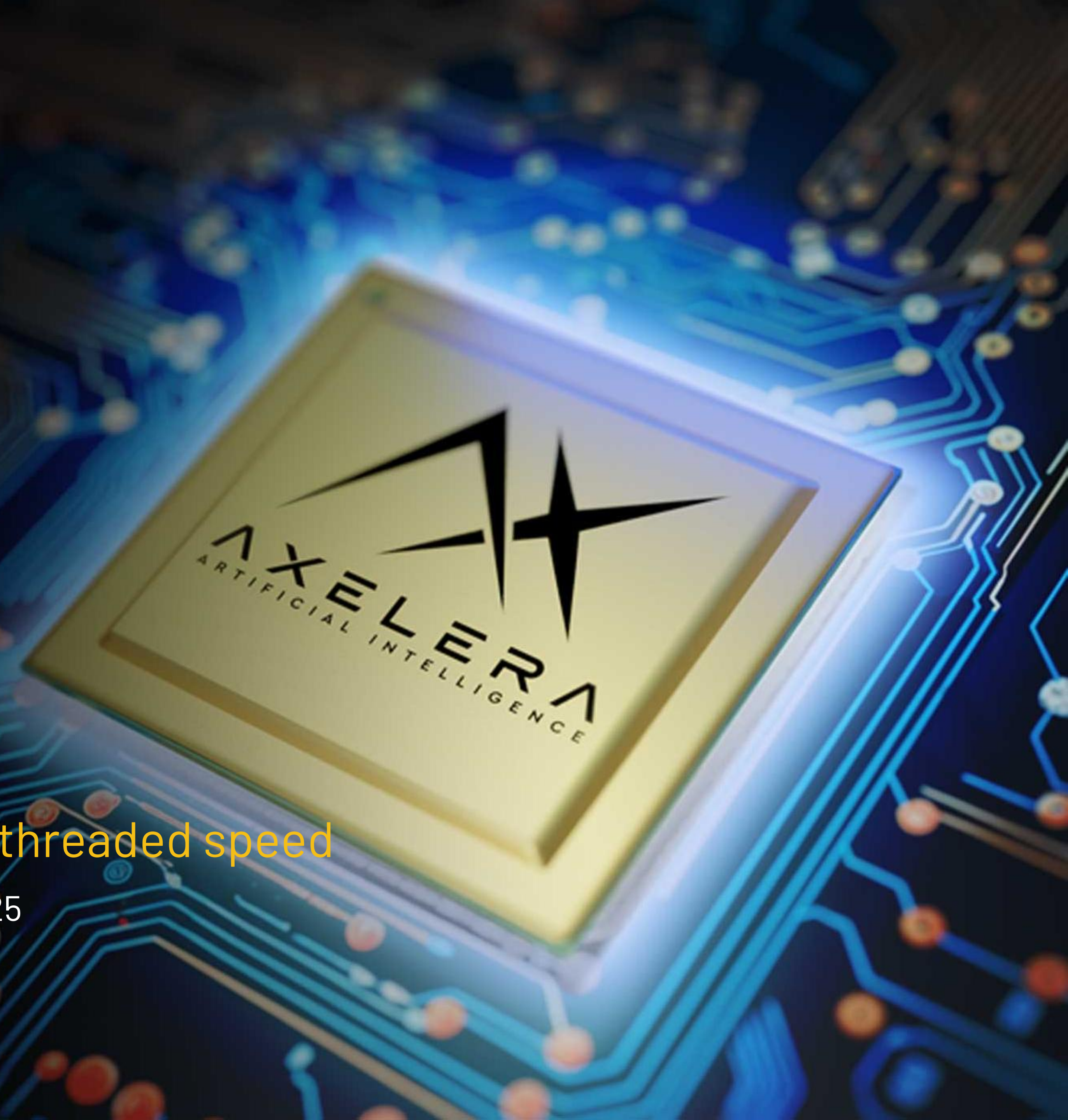
Datacenter Performance. Edge Efficiency.
Accelerating Inference, Everywhere.



Multisim: simulate RTL with real multi-threaded speed

Verification Futures Conference 2025 UK – 1st July 2025

Antoine Madec

Who am I?

- 14 years in simulation & emulation
- Europe and US
- Technical leader, FAE, manager: always hands-on
- Teams from 3 to 12 people
- Block-level / Top-level
- Automotive, Video Decoder, AI
- Axelera AI: accelerating inference

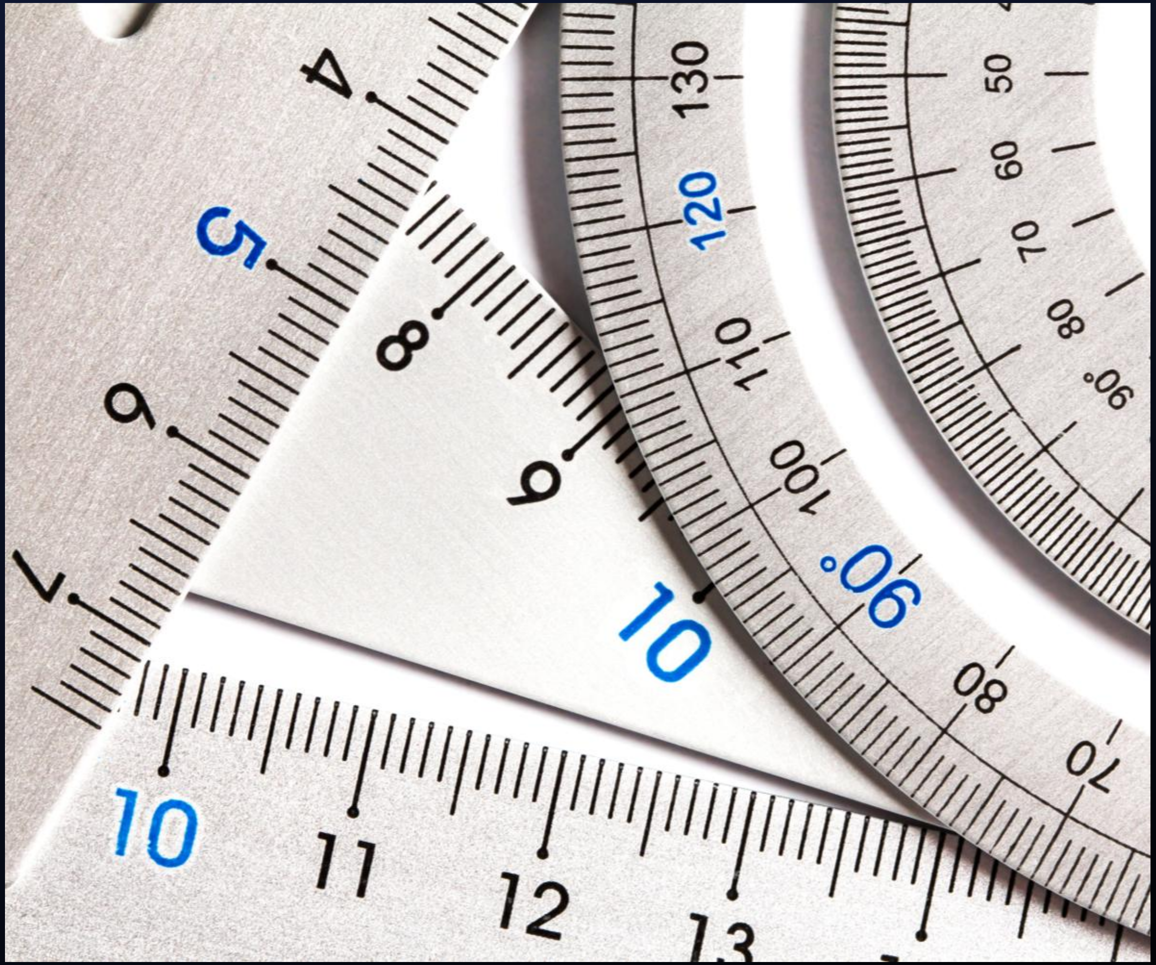
How it started

- AI and HPC chips are huge: how to simulate it in a reasonable time?
- Current EDA simulators speed don't scale nicely on big designs
- Need for scalability and "real" multi-threading performances

Trading cycle accuracy for speed

Is cycle accuracy always needed?

- Multi-threaded simulation can't be fast if cycle accuracy is needed
- Block-level and interfaces test benches can be tested separately
- Run a couple of basic/boot classic sim; run the rest with **multisim**
- **Cycle accuracy will be sacrificed**

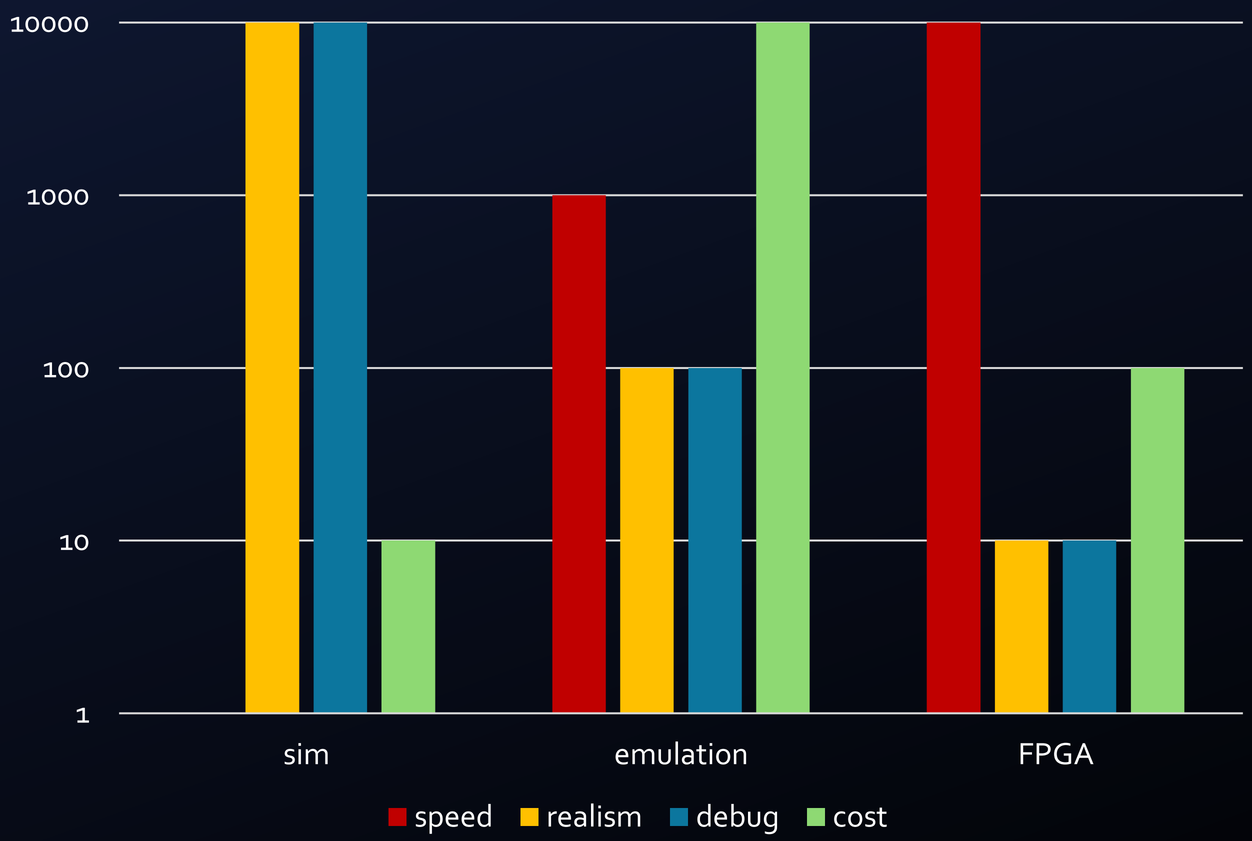


Verification platform landscape

Current verification platform landscape

- Big projects are too slow for classic simulation
  - Emulation, models, etc, are a must
  - The right platform for the right tests for the right interfaces
  - Trade-offs are inevitable
    - Speed
    - Realism
    - Debugability
    - Platform bring-up time
    - Price

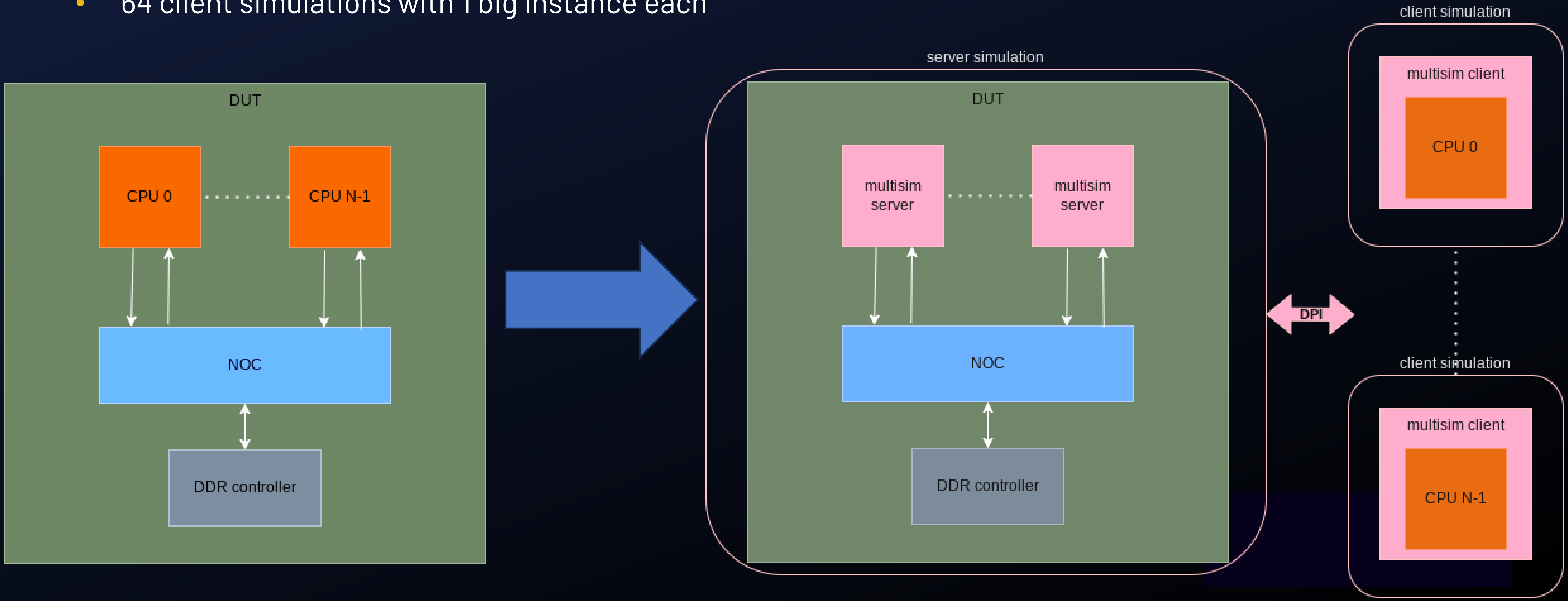
- It is our job to choose the right platform



Multisim: high-level overview

Multiple simulations running in parallel

- SV/DPI: uses TCP/IP to communicate
- 1 server simulation with your DUT skeleton (NOC, etc)
- 64 client simulations with 1 big instance each

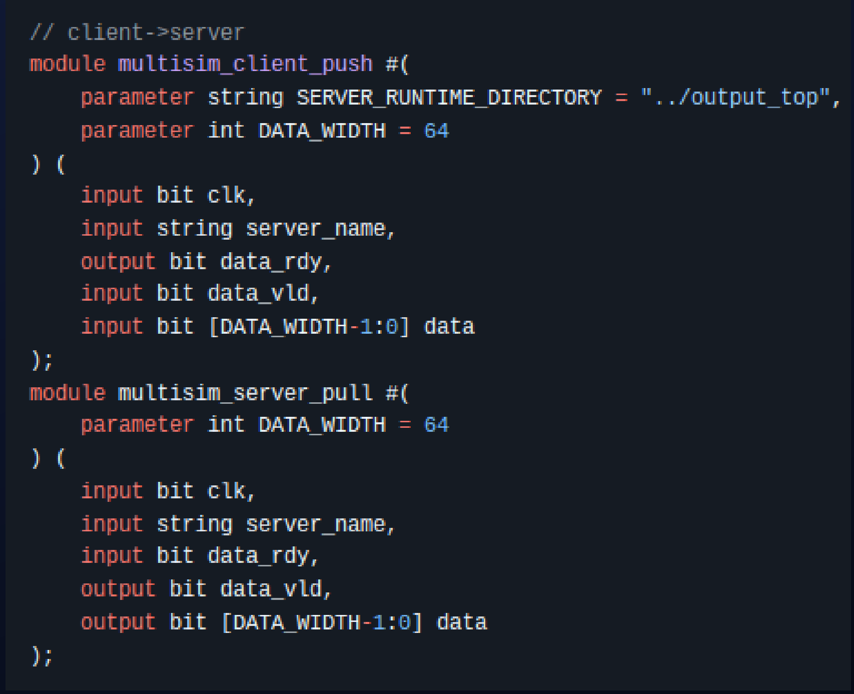


Multisim: low-level overview

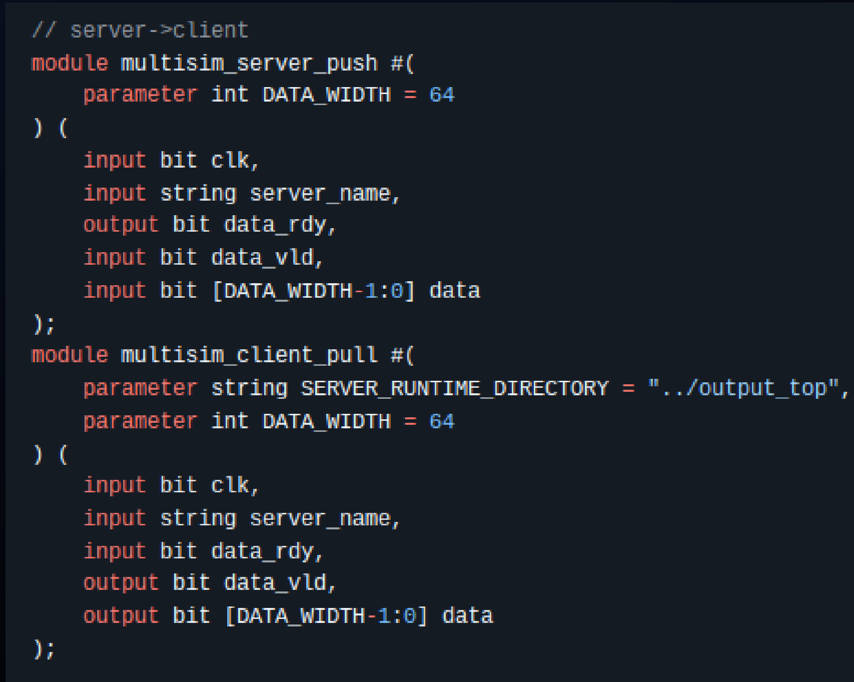
- Trade-offs and biases
  - API: simple
  - Complexity: abstracted from the user
  - Cycle accuracy will be sacrificed
- Channels
  - Server & Client communicate through channels (TCP/IP)

yes, but not in the presentation

  - Unique **server_name** to link a client/server channel together
  - Ready/Valid protocol to exchange DATA_WIDTH bits
  - Channels direction can be:
    - **client->server**: client_push() + server_pull()
    - **server->client**: client_pull() + server_push()



```
// client->server
module multisim_client_push #(
    parameter string SERVER_RUNTIME_DIRECTORY = "../output_top",
    parameter int DATA_WIDTH = 64
) (
    input bit clk,
    input string server_name,
    output bit data_rdy,
    input bit data_vld,
    input bit [DATA_WIDTH-1:0] data
);
module multisim_server_pull #(
    parameter int DATA_WIDTH = 64
) (
    input bit clk,
    input string server_name,
    input bit data_rdy,
    output bit data_vld,
    output bit [DATA_WIDTH-1:0] data
);
```

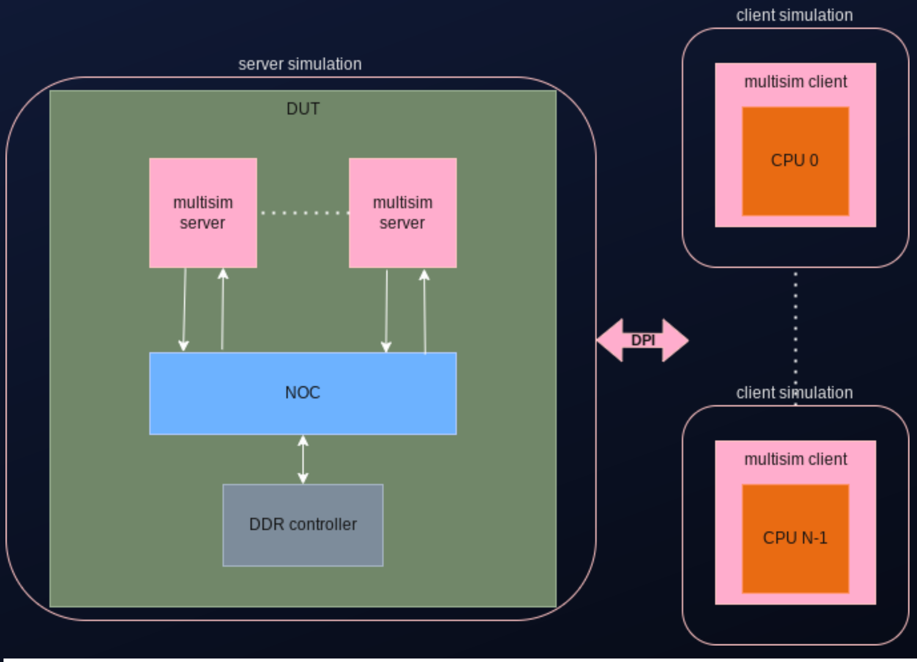


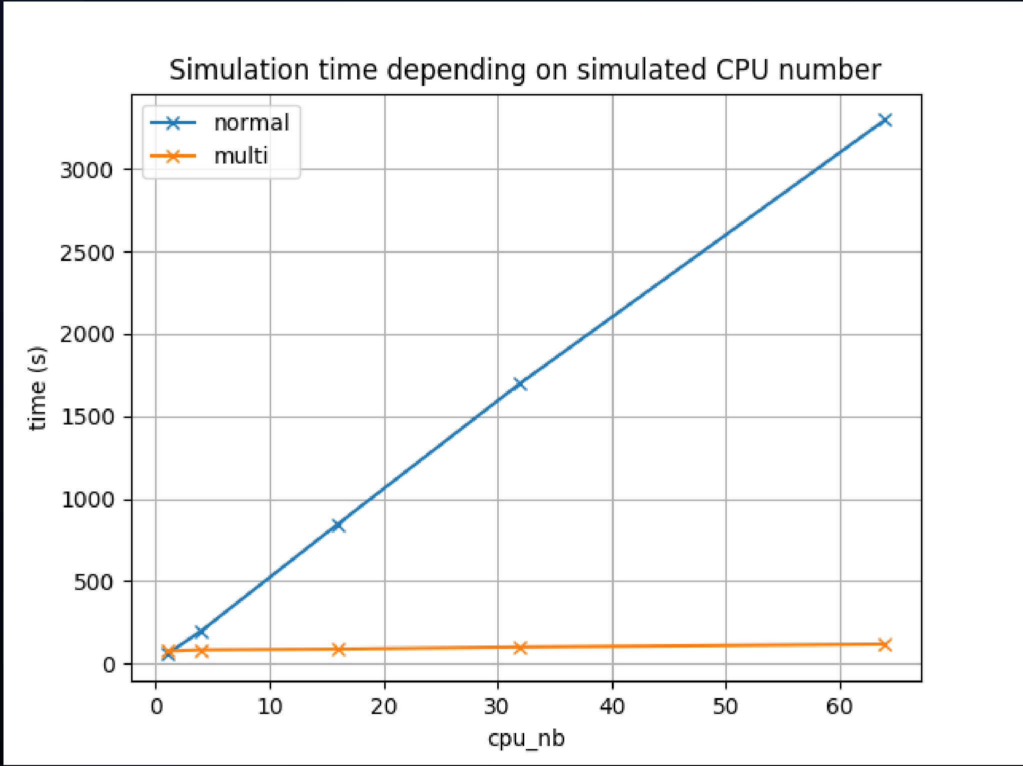
```
// server->client
module multisim_server_push #(
    parameter int DATA_WIDTH = 64
) (
    input bit clk,
    input string server_name,
    output bit data_rdy,
    input bit data_vld,
    input bit [DATA_WIDTH-1:0] data
);
module multisim_client_pull #(
    parameter string SERVER_RUNTIME_DIRECTORY = "../output_top",
    parameter int DATA_WIDTH = 64
) (
    input bit clk,
    input string server_name,
    input bit data_rdy,
    output bit data_vld,
    output bit [DATA_WIDTH-1:0] data
);
```

Performance

simulate your RTL with real multi-threaded speed

- Reusing [this example in the github repo]()
  - 1 server sim with 1 NOC
  - 64 client sims with 1 CPU
- TCP/IP sockets
  - 64 Clients: 2 sockets each (CPU->NOC, NOC->CPU)
  - 1 Server: 2*64 sockets
  - Total: 128 sockets for 65 sims
  - Scalable
- CPU threads/temperature
  - Main limiting factor
  - CPUs are cheap(er than emulator)

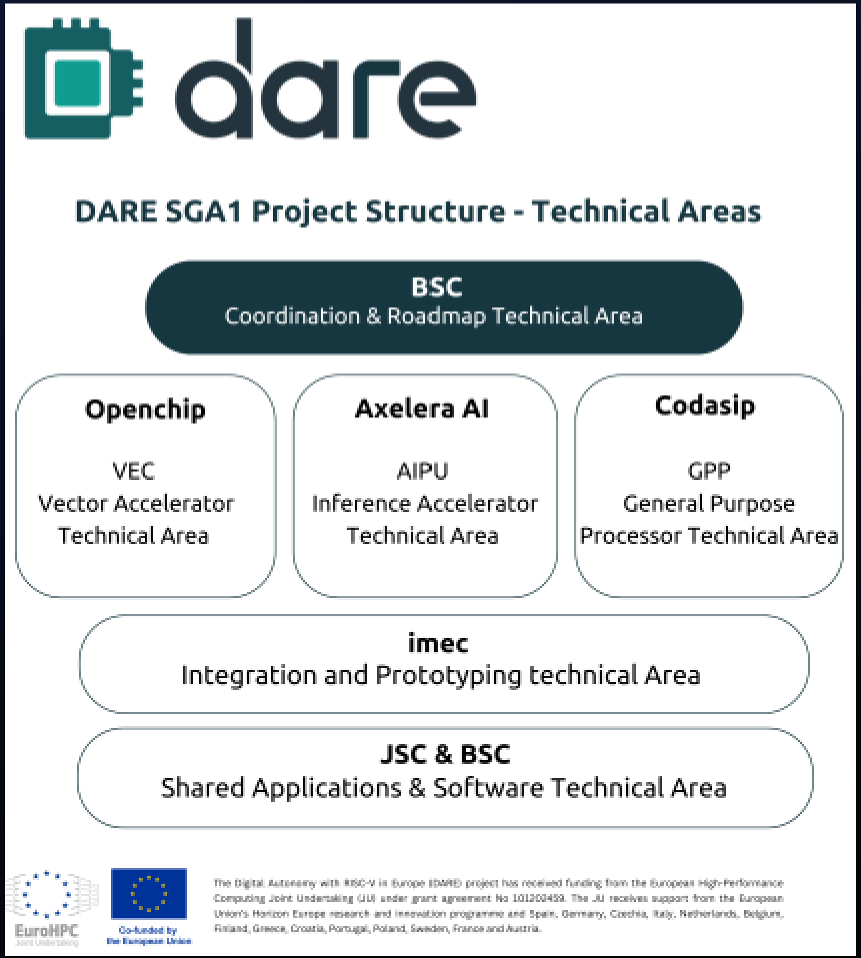




Interoperability

TCP/IP and DPI: the gate to freedom

- Multi company projects / chiplets
  - DARE (HPC): Openchip + Axelera AI + Codasip
  - Interfacing multiple simulation environments
    - Impose the same verif env and EDA to all companies?
    - Critical interfaces: the right tests at the right places (PCIe, etc)
    - Easily test the whole system with **multisim**
- Ideas
  - Share expensive resources / licenses with 1 server sim
  - Emulation: leverage complex sim models (slow?)
  - 1 EDA license for N Verilator simulators
- Constraints
  - Ping induces slowdown: run on the same server
  - Emulation: DPI access are slow


dare
DARE SGA1 Project Structure - Technical Areas
BSC
Coordination & Roadmap Technical Area
Openchip
VEC
Vector Accelerator Technical Area
Axelera AI
AIPU
Inference Accelerator Technical Area
Codasip
GPP
General Purpose Processor Technical Area
imec
Integration and Prototyping technical Area
JSC & BSC
Shared Applications & Software Technical Area
EuroHPC Joint Undertaking
Co-funded by the European Union

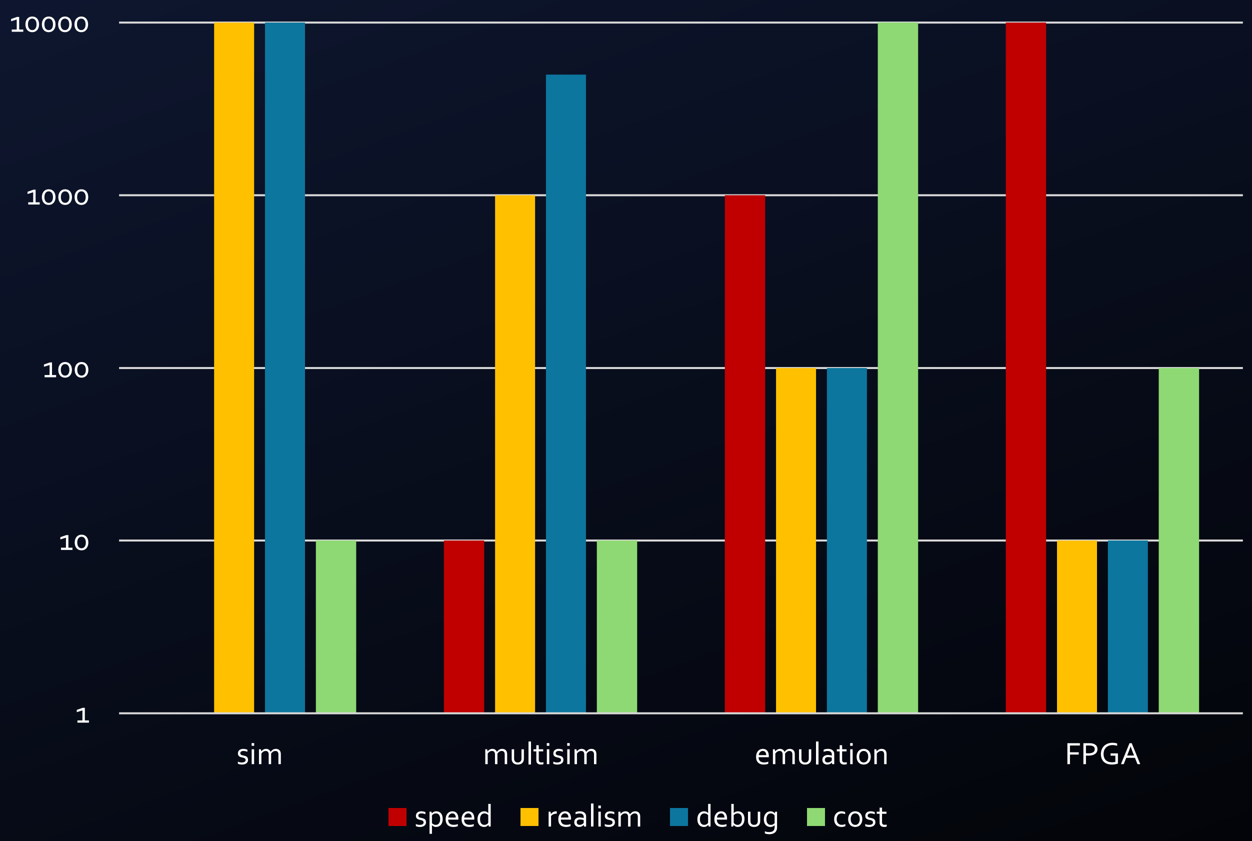
The Digital Autonomy with RISC-V in Europe (DARE) project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101202459. The JU receives support from the European Union's Horizon Europe research and innovation programme and Spain, Germany, Czechia, Italy, Netherlands, Belgium, Finland, Greece, Croatia, Portugal, Poland, Sweden, France and Austria.

Verification platform landscape (2)

Where does multisim fit in?

- Big projects are too slow for classic simulation
  - Emulation, models, etc, are a must
  - The right platform for the right tests for the right interfaces
  - Tradeoffs are inevitable
    - Speed
    - Realism
    - Debugability
    - Platform bringup time
    - Price

- It is our job to choose the right platform



Pros and cons

Multisim strenghts and limitations

- Pros
  - **Scalability**: as long as you have enough CPUs on your server
  - **Speed**: split your big DUT in as many smaller parts as you want
  - **Cost**: server CPUs are cheaper than emulation solutions
  - **Bringup** time: easy SV modules, simple interface (e.g.: AXI is 5 channels)
  - **Interoperability**: each server/client can use different simulators (Verilator, VCS, Questa, Xcelium, etc)
    - Supports different compile flows
    - GLS with RTL, etc
- Cons
  - **Not Cycle Accurate**: functionally accurate, but not cycle accurate
  - **Harder Debug**: waveforms split on N+1 simulation, no time coherency in between them

Future improvements

- Repo: https://github.com/antoinemadec/multisim

- Future improvements
  - higher level of abstraction
    - e.g.: AXI is 5 channels
  - simple transaction logging to help debug
  - ...

AXELERA
ARTIFICIAL INTELLIGENCE

Thank You!