2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
JAPAN

High-Speed Emulation Framework for Performance Analysis of GenAI SoC design

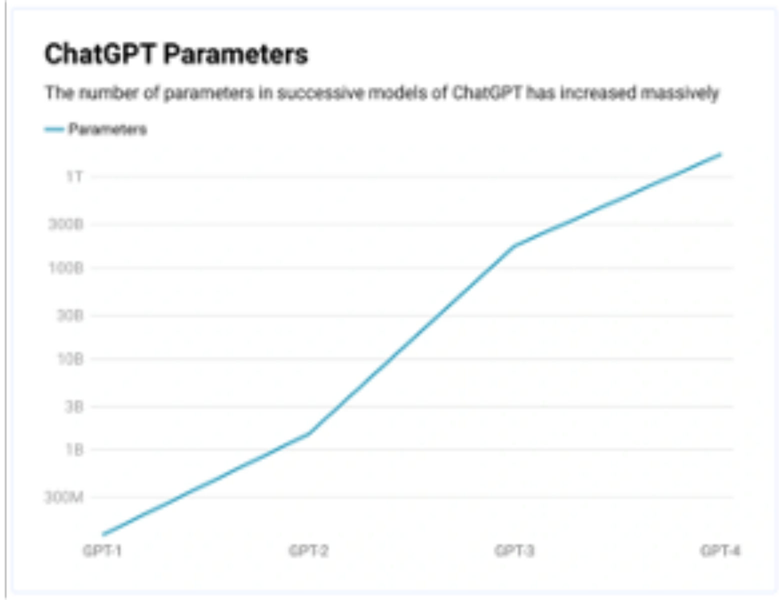
Abhishek Saksena, Kalyan Kar, Saksham Mehra

Google

accellera
SYSTEMS INITIATIVE

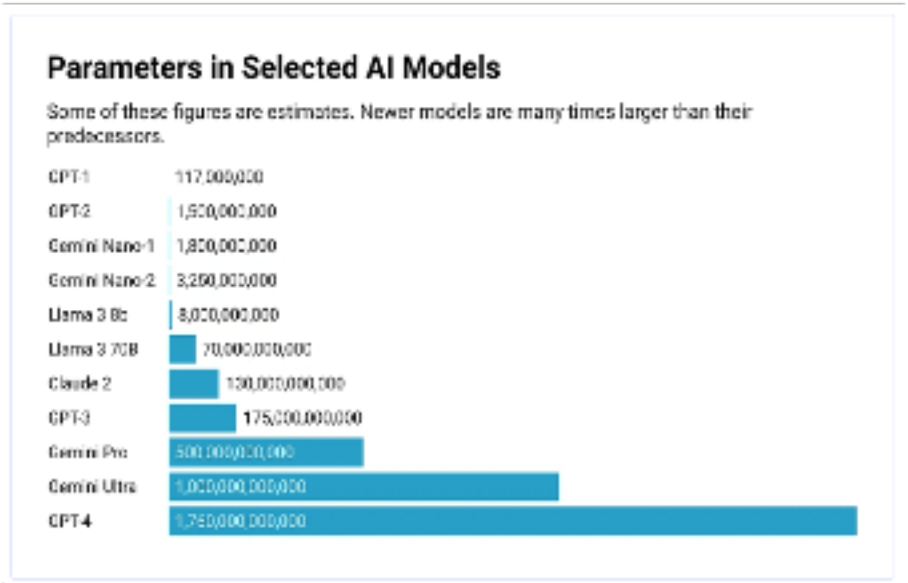
GenAI: Model and Workload Growth Trajectory

Number of parameters in genAI model are increasing exponentially



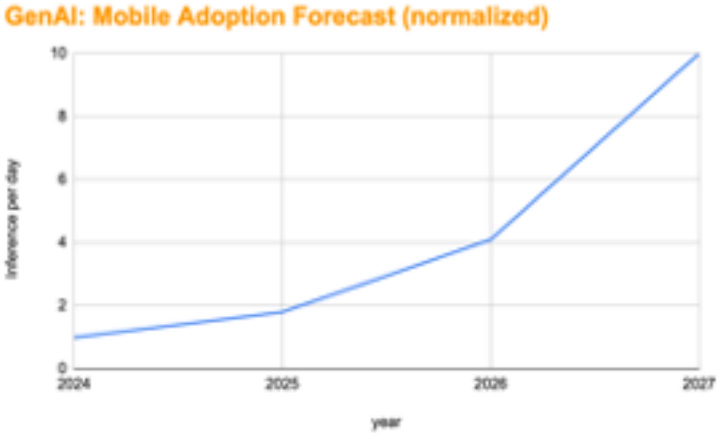
ChatGPT Parameters

The number of parameters in successive models of ChatGPT has increased massively



Parameters in Selected AI Models

Some of these figures are estimates. Newer models are many times larger than their predecessors.



GenAI: Mobile Adoption Forecast (normalized)

- More parameters necessitate a proportional increase in DRAM bandwidth.
- Higher daily usage requires improved power efficiency.
- Rigorous pre-silicon verification is essential to ensure performance and power targets are met.

Traditional Platforms for SoC performance testing

SoC Perf C Model

Platform availability : Very Early in design cycle

- Can run Synthetic tests for latency & BW check
- Benchmark/use case traces can also be run
- Simulation speed - decent.
- Perf numbers might be little different from actual RTL/silicon run. Correlation with RTL results needed
- Ideal for usecase/benchmarks testing as well as synthetic tests

SoC RTL Simulation

Platform availability : Early in design cycle

- Can run Synthetic tests for latency & BW check
- Full Use-case & Benchmark run - not feasible. Takes several days to complete
- Simulation speed - slow.
- Performance numbers close to silicon
- Debug visibility: high

SoC RTL Emulation

Platform availability : Late (After production firmware bringup)

- Synthetic tests for latency & BW check – not easy
- Benchmark/Use-case can be run easily
- Simulation speed - very fast.
- Performance numbers very close to silicon
- Debug visibility: limited
- Ideal for usecase/benchmarks testing

Need for a ‘new’ platform

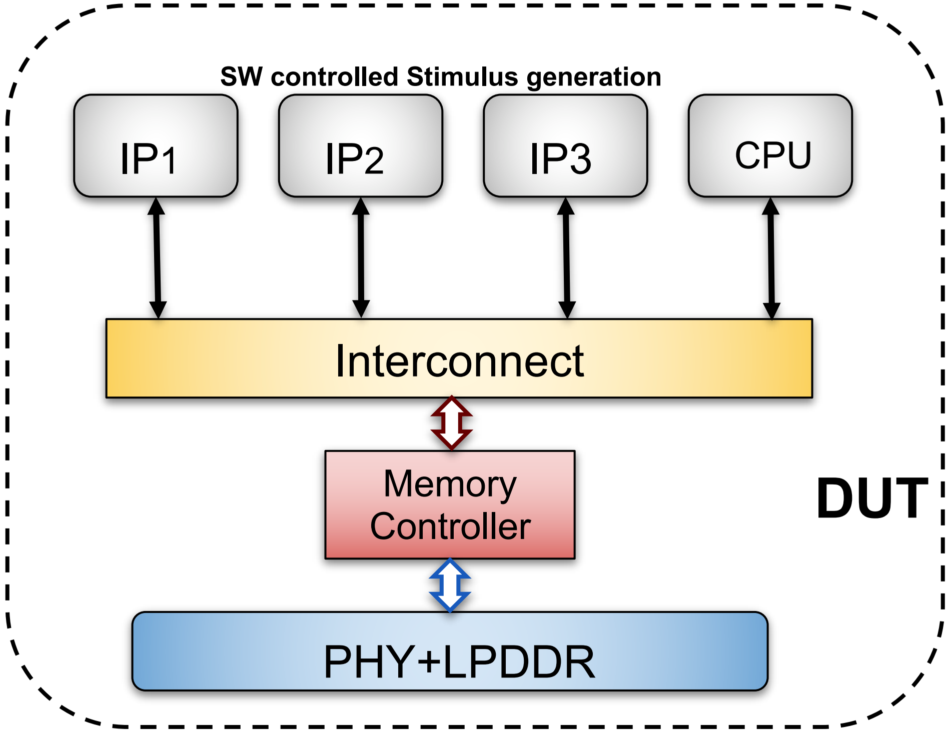
Issues with traditional SOC emulation testbench ?

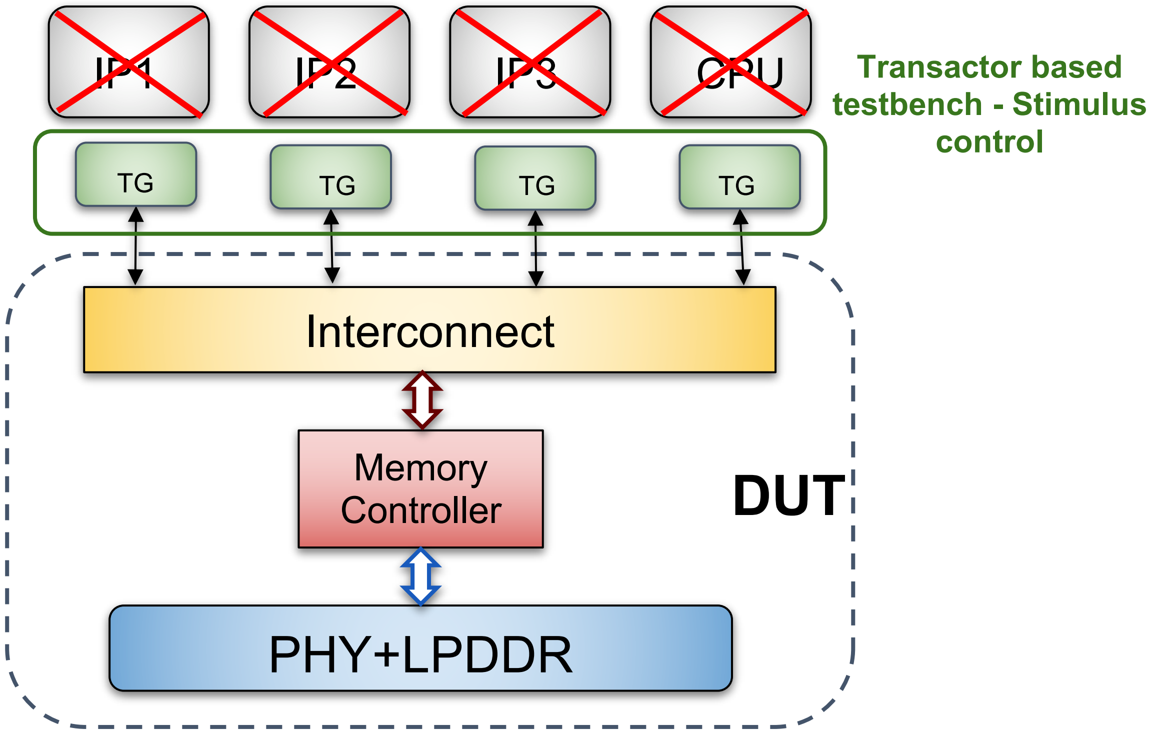
- **Importance of Benchmark execution: Crucial for evaluating SoC design performance.**
- **Need for Emulation Platform: Required for stress testing performance use cases early in the project cycle without software dependency.**
- **Left-Shift Requirement: Early emulation helps find critical bugs and enables architectural explorations, allowing feedback for re-synthesis within the same project cycle.**
- **Cost-Effectiveness: Emulation capacity are scarce and expensive; solutions must be cost-effective to optimize their usage.**
- **Stimulus Control and Accuracy: Solutions should offer easy stimulus control and be cycle accurate for testing synthetic pattern**

Solution

- **Lite Uncore Emulation model: Introduce an 'uncore' emulation model.**
- **Uncore SoC model Components: Includes interconnects, memory controller, and DRAM/PHY blocks, with compute and multimedia cores stubbed.**
- **Independence from CPU Bootup: No dependency on CPU bootup or production firmware availability.**
- **Traffic Generator Integration: Add a traffic generator at IP initiator ports to create synthetic and use case traffic without booting up the cores.**

Proposed Testbench Structure

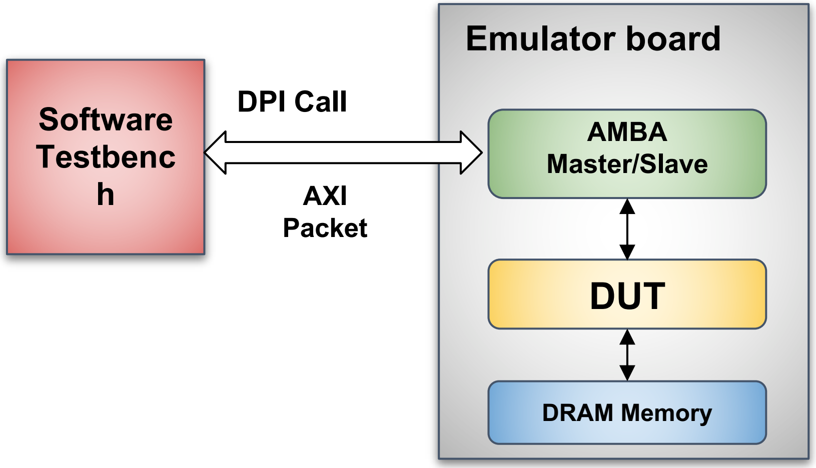




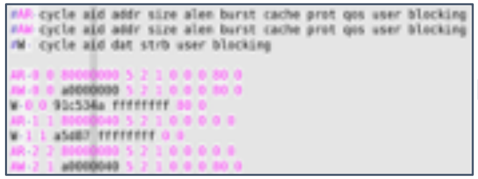
- **Easier Stimulus Control**: Traffic generators reduce software overhead and dependence. Offers finer stress control.
- **Integration of Monitors and Loggers**: Simplifies debugging in the 'lite' testbench
- **Reduced Emulator Capacity**: Major compute cores are stubbed out, occupying less emulator resource; improving overall capacity utilization for all users
- **Improved simulation speed**: High speed run due to synthesizable stimuli generator; no emulation stalling due to software-based verification collaterals

Traffic Generator

Hybrid AXI Transactor

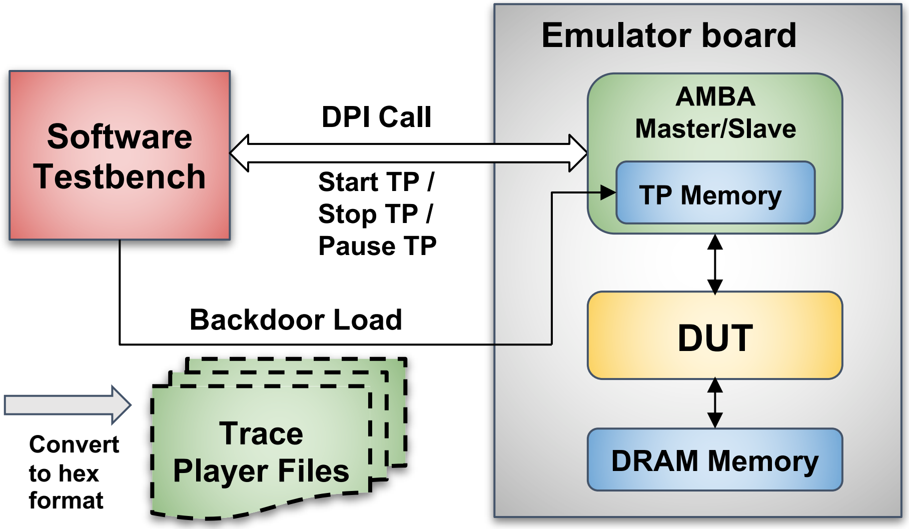


Fully synthesizable trace player



```
#AR cycle aid addr size alen burst cache prot qos user blocking
#AW cycle aid addr size alen burst cache prot qos user blocking
#W  cycle aid dat strb user blocking

AR 0 0 00000000 5 2 1 0 0 0 00 0
AW 0 0 a0000000 5 2 1 0 0 0 00 0
W 0 0 91c534a ffffffff 00 0
AR 1 1 00000040 5 2 1 0 0 0 0 0
W 1 1 a5d87 ffffffff 0 0
AR 2 2 00000000 5 2 1 0 0 0 0 0
AW 2 1 a0000040 5 2 1 0 0 0 00 0
```

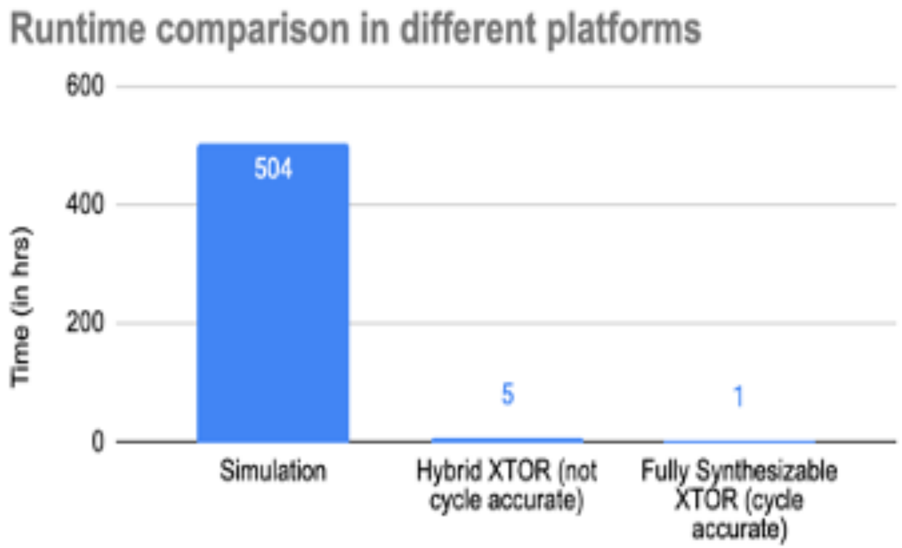


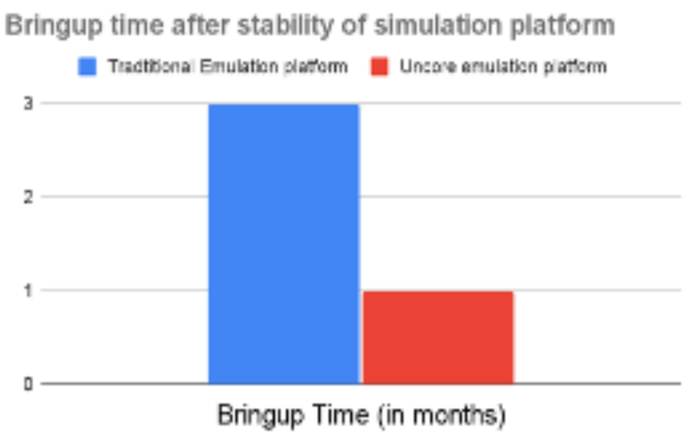
- **Lack of Cycle Accuracy:** Transactions are driven into the DUT indeterministically and lack timing accuracy, not ideal for performance evaluation
- **High Software Overhead**: Enabling cycle accurate timing mode requires syncing after every transaction, stopping/stalling the design clock. Severely degrading emulation utilization.
- **Expensive DPI Calls**: In benchmarks with millions of transactions, overall test runtime suffers due to DPI calls bottleneck.

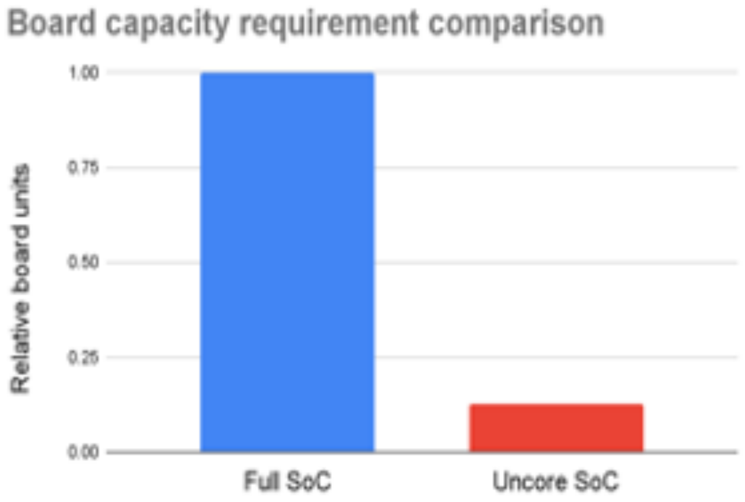
- **Trace Player Synthesis**: Synthesizes complete trace player along with the SoC design
- **Trace loading Support:** Transactions and delay information are converted to a hex file and loaded into board memory and played back with 100% timing accuracy.
- **Transaction Playback and Control**: Trace player can be started, stopped, or paused using API, playing traffic in a cycle accurate manner
- **On board Memory for Emulation Platforms**: Industry standard platforms like ZeBu have enough memory on board to accumulate large use case trace files (e.g., 1M transaction trace fits in 16 MB memory)

Results / Findings

- **Use Case Performance Verification**: Ideal for long running GenAI benchmarks which involves millions of memory transactions . Full performance usecase verification completed on SoC RTL well before design freeze, providing feedback to architects within the same project cycle.
- **Additional Applications:** MMU cache sizing studies, ROB depth analysis, Tuning for QoS performance settings for architectural explorations leading to next generation designs with better PPA.


Runtime comparison in different platforms
Time (in hrs)
Simulation
Hybrid XTOR (not cycle accurate)
Fully Synthesizable XTOR (cycle accurate)


Bringup time after stability of simulation platform
Traditional Emulation platform
Uncore emulation platform
Bringup Time (in months)


Board capacity requirement comparison
Relative board units
Full SoC
Uncore SoC