

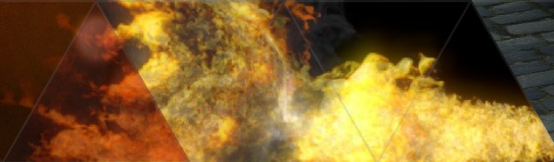
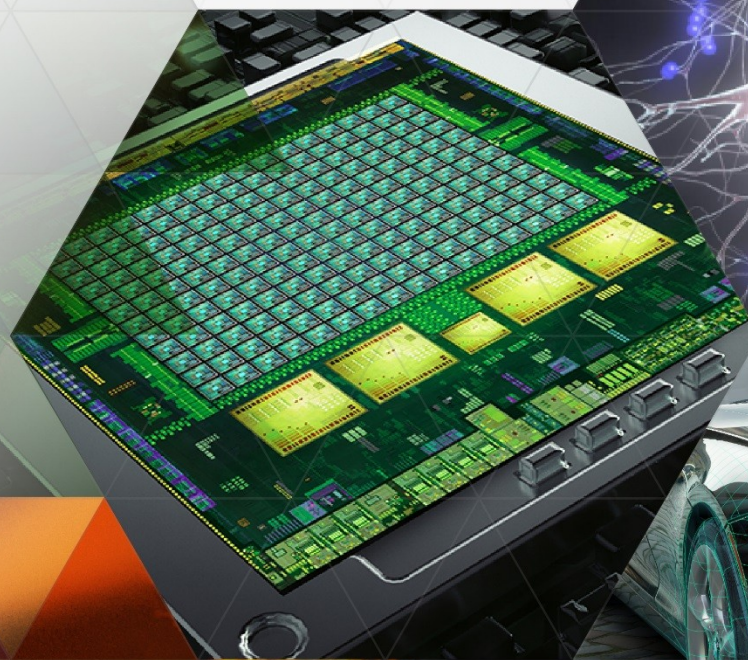


MODSIM'15 INDUSTRIAL PANEL

Alex Ramirez

Principal Research Scientist

NVIDIA Research



HOW DO WE USE THE TOOLS?

Internal, external / community, or both?

- ▶ *No one tool to rule them all*
 - Different tools at different stages
 - Different tools for different metrics
 - Internal tools as we advance in the design
- ▶ Community can greatly contribute
 - Representative workloads
 - New ideas and features to tackle fundamental problems
 - Possibly work together using top-level tools

BEWARE OF SIMULATORS

Fix real problems vs. Fix the tool

- ▶ Focus on fundamental architecture issues
 - Avoid fixing simulator quirks
 - Don't try to *teach daddy how babies are made*
- ▶ The danger of *validated* simulators
 - Validated against what? An outdated product?!
 - How can you validate a new feature?
- ▶ Easy to rely on not-available information
- ▶ Easy to ignore complexities

WHAT MAKES A TOOL USEFUL?

Keys to success

- ▶ Provide INSIGHT vs. lots of aggregated results
 - Care about direct impact metrics, not speedup
- ▶ Feature rich, flexible and extensible
- ▶ Speed
 - Ability to run representative workloads and systems
- ▶ Fidelity to the architecture
 - Not to confuse with accuracy / validation

PERFORMANCE, POWER, RELIABILITY

All of them, of course

- ▶ Not one tool to rule them all
 - Can't simulate full system in full detail
 - Evaluate separately, compose metrics
- ▶ Workload scaling much harder than system scaling
 - Input sets, moving bottlenecks
- ▶ Reliability is a multi-layer problem
 - Circuits, architecture, runtime, system, application, ...

WHAT ARE THE CURRENT GAPS?

How can the community help?

- ▶ Lack of *representative* workloads
 - Large enough to stress full system
 - Main memory, network, filesystem
- ▶ Miniapps tend to use *random* data
- ▶ Insight on application requirements
- ▶ It is very hard to simulate a workload that does not exist
 - Exploiting new architecture features
 - Support for new/upcoming programming models