Using GPUs to Speedup Chip Verification

Tomer Ben-David, VP R&D
The Verification Bottleneck

- Long verification process
  - 66% of designs, verification takes 50% of the design cycle
  - In ~40% of projects, simulation regression runtime is longer than 1 day

- SoC simulation challenge
  - Over 40% of designs are larger than 10M gates
  - Difficult to simulate the entire design/SoC

- Excessive computing resources
  - Required 10’s or 100’s GBytes of memory
  - Needed most advanced CPUs

Source: 2010 Wilson Research Group and Mentor Graphics Functional Verification Study
• Verification is the bottleneck of the design cycle

• Simulation speed is the bottleneck of verification

• RocketSim\textsuperscript{TM} accelerates simulations by 10X or more
Simulators using CPUs

• “Event driven”, implemented with a single queue of events
  – HDL (Verilog / VHDL) naturally maps to “event driven” even better than it maps for “synthesis”
  – Very short-simple calculations, handling them one-at-a-time makes a lot of sense

• Memory access patterns
  – For every complete clk-cycle, most of the simulation state is accessed
  – As design size increases we get more cache miss

• Multi-core CPUs:
  – The HW
    • Only one order of magnitude
    • Limited memory bandwidth
  – The SW solutions
    • High-level partitioning
    • Relaxation latency
Applications using GPUs

Source: NVIDIA
Typical CUDA development flow

1. Application/Problem
2. Find algorithm that solves the problem
3. Split code to CPU/GPU regions
4. Coding for CPU/GPU parts
5. Debug & fix
6. Profiling and Performance analysis
Many CUDA developers are facing the same challenge:

→ Efficient coding is not trivial

In our case the “The Problem” varies with every User’s chip design:

→ Graph with billions of processing nodes
→ Huge amount of dependencies
→ Mapping different Verilog constructs to SIMT
→ Compilation of User’s design has to take minutes, development flow with CUDA takes weeks!
The solution: “Virtual Machine”

- Rocketick’s VM is implemented using CUDA
- The VM processes “recipes” efficiently, maintaining graph dependencies
- Proprietary tools for debug and profiling
- Ideal-fit target “platform” for RocketSim’s compiler
- Each GPU forward compatibility
Rocketick’s technology
“Breaking the Dependency Barrier”

Complex dependency graph

Rocketick’s algorithms

DRAM

Rocketick’s Virtual Machine, Implemented in the GPU
Traditional Simulators Vs. RocketSim

Compilation

Design + Test-bench Sources

Host Simulator Compiler

User PLI Modules

Host Simulator (NC-Sim, VCS, ModelSim)

VCD/FSDB

Logs

Compilation

Design + Test-bench Sources

RocketSim Compiler

Accelerated modules (RocketSim proprietary)

Non-accelerated modules (RocketSim re-generated)

Host Simulator Compiler

User PLI Modules

Host Simulator (NC-Sim, VCS, ModelSim)

PLI Module

GPU

RocketSim Offload Engine

VCD/FSDB

Logs
RocketSim - Compilation Stages

**Analyze**
- Parse HDL source files
- Static Elaboration
- RTL Elaboration

**Compile**
- Create optimal dependency graphs
- Calculate optimal GPU invocation schemes
- Generate skeleton (ske.v)

**Assembly**
- Calculate optimal memory allocation for variables
- Generate final recipes for the GPU virtual machine
The challenge of squeezing out performance

- Wide memory accesses
- Efficient Kernel invocations (static & dynamic work aggregation)
- Very large dataset, very small GPU shared memory
- Host-GPU synchronization (simulation state)
- Long sequential logic (implementation optimizations and GPU-CPU handoff)
- Multi-GPUs (much larger latencies)
Nvidia tools Vs. proprietary tools

- When “gdb” is just not enough –

- When “CUDA profiler” is just not enough –
RocketSim™ bringing results!

“With RocketSim, ATPG gate level simulation time is reduced dramatically from weeks to days, with a tenfold increase in speed and five-fold decrease in server RAM requirements.”

Jonah Alben
Senior VP of GPU Engineering, NVIDIA
RocketSim™ Overview Summary

» 10x or more acceleration factor

» Works seamlessly *with all leading simulators*

» Supports extremely huge designs *(Giga-gates)*
Rocketick

Need

• Big industry pain - Huge amount of $’s spent on verification

Challenge

• Mapping graph with dependencies to GPU – very hard to do!

Results

• 10X-30X acceleration on the largest designs in the word
Thank you

for more information please visit our web site:

www.rocketick.com