The Need for Speed: Understanding design factors that make multi-core parallel simulations efficient

Shobana Sudhakar
shobana_sudhakar@mentor.com

Rohit K Jain
rohit_jain@mentor.com

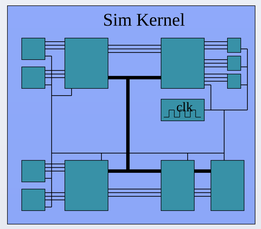
Mentor Graphics

Introduction

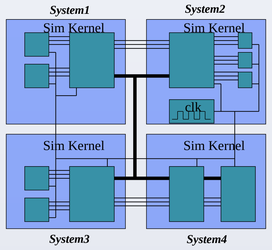
Multi-core parallel simulations work very well on designs that meet certain criteria and these factors help to

- Qualify the design for multi-core simulations
- Design multi-core-friendly designs

Single-core simulation



Multi-core parallel simulation



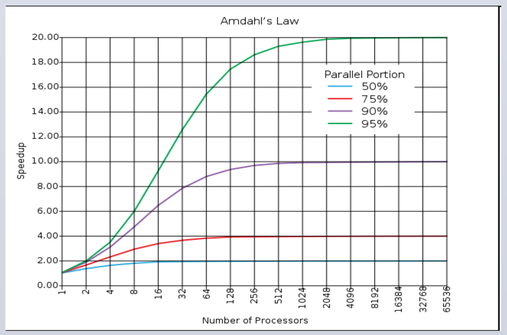
Motivation

Questions raised by customers during QuestaSim MC^2 (Multi-core Multi-computer) technology deployments:

- Do multi-core/parallel simulations even work?
- When do parallel simulations work? When do they not work well?
- What can be done to make it work?
- What should be used to speed-up existing regression suites – a distributed grid job or parallel simulations?

Maximum speed-up possible on multi-core simulations

$$\frac{1}{(1 - P) + P/N}$$



Amdahl's Law dictates the upper limit on speedups that can be achieved through multi-core simulations depending on the fraction of parallel portion

Design factors that affect multi-core parallel simulations performance

Design factors discussed here are not restricted to any single EDA simulator but provides guidelines to understand what makes multi-core simulations a success.

Load balancing:

Very good load balancing inside the DUT

```
# testbench     100%
#   dut         91.1%
#       block1   32.6%
#       block2   22.7%
#       block3   22.6%
#       block4   21.0%
```

Very uneven load balancing with very low DUT time and possibly sequential high-level monitor/drivers

```
# testbench    44.9
#   dut         6.3
#       interface   5.5
# package      15.7
# monitor      14.0
# driver        8.8
```

Concurrency:

Designs such as multi-core SoC with busy cores and independent functions → good candidates

Partition p3 is idle most of the time and this diagnostic report indicates sequential execution of design.

	p0	p1	p2	p3
Total Time	37.49s	37.49s	37.49s	37.49s
Idle Time	0.00s	2.29s	0.54s	27.00s

Balanced blocks that are not concurrent and active one after the other → bad candidates

Communication:

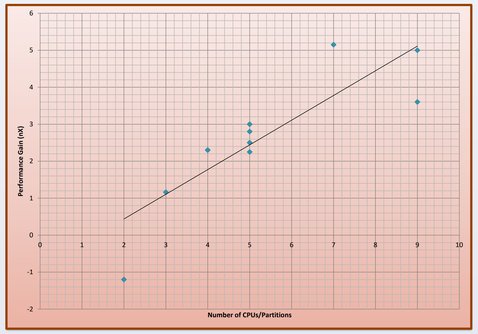
Heavy inter-partition communication (IPC) for data transfers and synchronization → negative impact on multi-core simulations.

✖ Designs with flat hierarchy or partitioning at a lower hierarchy -- many communication ports, increased traffic

✔ Blocks that communicate a lot → keep together in single partition

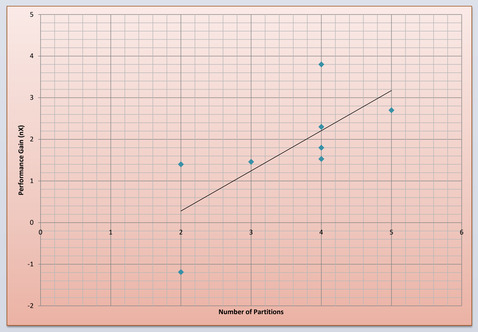
Results

Results of RTL customer designs charting various performance gains vs. number of parallel partitions used



- Performance gain with parallel simulations – 1.2x slower to 5.15x faster
- ✔ Balanced and parallel activity in partitions
- ✖ Heavy class-based TB and little to no functional activity in the DUT

Results of Gate Level customer designs charting various performance gains vs. number of parallel partitions used



- Performance gain with parallel simulations – 1.19x slower to 3.8x faster
- ✔ Design activity in gate-level simulations typically tend to be more balanced after synthesis
- ✖ Unbalanced simulations with majority of time spent in a small block

Commercial multi-core simulation provider checklist

- Flexible methods to create design partitions – manual, semi-automatic or fully automatic
- Support all HDL languages and language features
- Partition design based on dynamic simulation activity
- Control cross partition synchronization flexibly
- Various analysis reports to provide feedback on design suitability for multi-core simulations as well as performance results
- Fewer number of tool limitations that affect partitioning

Applicability of parallel simulations technology

Design qualification criteria which can help lead to successful multi-core simulations:

- Big designs with long simulation times of more than an hour
- Balanced activity in each partition
- Flat gate-level netlists and designs with little to no hierarchies do not partition well and are not good candidates
- Minimal cross-partition access such as through PLI/DPI/FLI/VPI usage
- Race-free designs

Regression suite of tests:

- ✖ Large number of tests with small simulation time
  - Multiple simulation jobs in parallel to a distributed grid
- ✔ Tests that take multiple hours/days
  - Faster identification and fix of functional issues with shorter design cycle

References

Results and analysis reports from actual evaluations and deployments of QuestaSim MC^2 technology on real customer designs