2015 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

A Methodology to Port a Complex Multi-Language Design and Testbench for Simulation Acceleration

Horace Chan, PMC

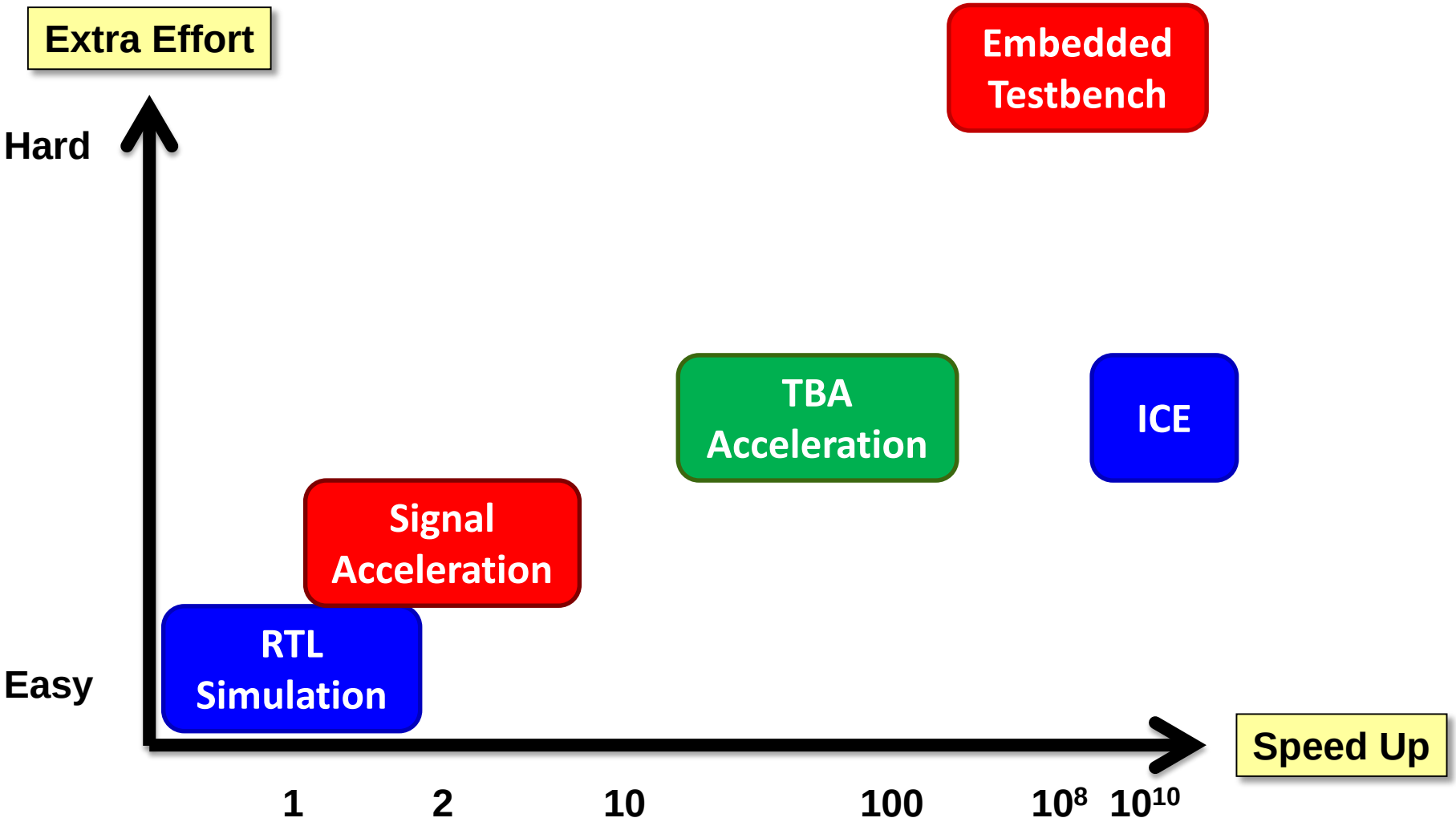
Brian Vandegriend, PMC

Efrat Shneydor, Cadence

cadence™

PMC

Why Simulation Acceleration



Benchmark Results

	Project A (2012)	Project B (2014)	Project C (2014)
Design size (in Mgates)	26	38	50
Domains used	6	6	8
Speedup factor	40x	67x	52x
Migration Schedule	> 1 year	3 months	3 weeks

4 Types of Simulations

Type	Where the RTL runs	Objectives
Normal	RTL Simulator	TB profiling, optimization Generate golden reference
SA_SIM	RTL Simulator	Debug SCEMI-pipe Debug DPI integration Clean up TB-RTL binding
SA_SW	SW model of the Box	Debug RTL synthesis flow Setup shell scripts for SA
SA_HW	The Box	Debug mismatch between SW model and the Box Benchmark SA

Compile the RTL

If the RTL won't compile, there is zero speedup

- Every VHDL/Verilog compiler is not the same
- Use the same tool chain from the same vendor
- Don't hack the RTL to work around tool problems
  - **fix the tool!**
- Use the same workspace as normal simulation
- Reuse compile scripts from ICE flow
- Use SV config to swap in TBA components

Testbench Enhancement

- Testbench is the bottleneck
  - Transaction based processing is a must
  - TB should use less than 2-3% CPU cycle
- Clean interface between TB and RTL
  - No wait delay
  - No direct RTL signal access
  - No backdoor access
  - No cycle by cycle interaction

Testbench Enhancement

Specman makes the TB upgrade easy

- Use AOP to keep TB code for normal simulation and acceleration in two different aspects
  - No messy factories or callback functions
- Use powerful macros and reflection interface to catch unknown TB-RTL interaction
  - Build debug tools to inspect the TB code

Regression Management

- Use the same regression tool as normal simulation
  - SA is like a super fast computer
- Download snapshot to the box is slow
  - Download once, run all the sims in the same session
- Auto re-launch overnight failed sims
  - SA_HW is too expensive for interactive debug
  - Hot swap to SA_SW before the failure point

- No checkpoint save and restore in SA (yet)

Conclusion

- Focus on quick bring up of SA
- Don’t optimize for the best possible speed up
  - SA can never run as fast as ICE
- Positive ROI around 30-40x speed up*
  - In terms of raw simulation through put

* depends on the license fees discount and acceleration box lease contract

- Faster debug turnaround time = higher productivity