# PANEL : The Future of SystemC

# Is there one?

- “Why can’t we just use QEMU”
- “My EDA Vendor gives me a model, I don’t care what it uses under the hood”
- “Its so slow”

- “What is SystemC anyway?”

# Whos' driving?

Is it fit for purpose? Whos' trying to make it fit for purpose.

- EDA vendors?

- S/W people – not even in the room?

- Are you in enough pain to contribute and make a difference?

# Live vote

- Hands up :
  - Are you as a user interested in contributing to the SystemC?
  - Would your company support contribution ?
  - Which area would you be interested in contributing to?
    - Next Generation Kernel ? (host multi core . . . )
    - Interface definition ? SoC interface / Interchip interfaces
    - Content: Models Cores / Peripherals etc

# What do you use SystemC for

- Hands up :
  - Are you LT or AT
  - HLS
  - AMS
  - IP reference models (verification etc)
  - Software engineer using SystemC
  - Model writers
  - SoC Architects


# Process : is it alive?

- Align our process with C++ -- 3 year release cycle
  - Year 1 : Proposals/pull requests to a public repository
  - Year 2 : Refinement of proposals within the WG
  - Year 3 : Publishing to the IEEE.
- Why not “in parallel” – Because it’s always the same people!
- Proposals should, where possible, be driven through working groups
  - ’SCP’ for instance can be used as a ‘sandbox’ in which to experiment. (e.g. ‘reporting’).
  - Available publically immediately.

# What should be where?

- user code
- Framework – is this SCP?
  - But we want items from here to be ‘composable’ with external items
  - So the ‘framekwork’ is needed in the standard?
- Associated libraries (are these standard or not?)
- SystemC core

# Big rocks that are missing

- TLM-2.0 is too old, and too painful.
  - Should be cleaned up (brought up to date with c++17 for instance, remove anoying template arguments for instance, . . .)
  - SERIAL PROTOCOLS – Please !
- CCI needs a lot of rework (see discussion later)
- "Framework" in the standard:
  - Reporting library needs to be made usable
- External simulation connections
  - 'cloud based' tlm …
  - other forms of federation
- More general Parallelism.
- **Everything in Python**

# EYCK JENTZSCH : ALTERIS / MINRES

# Future of SystemC

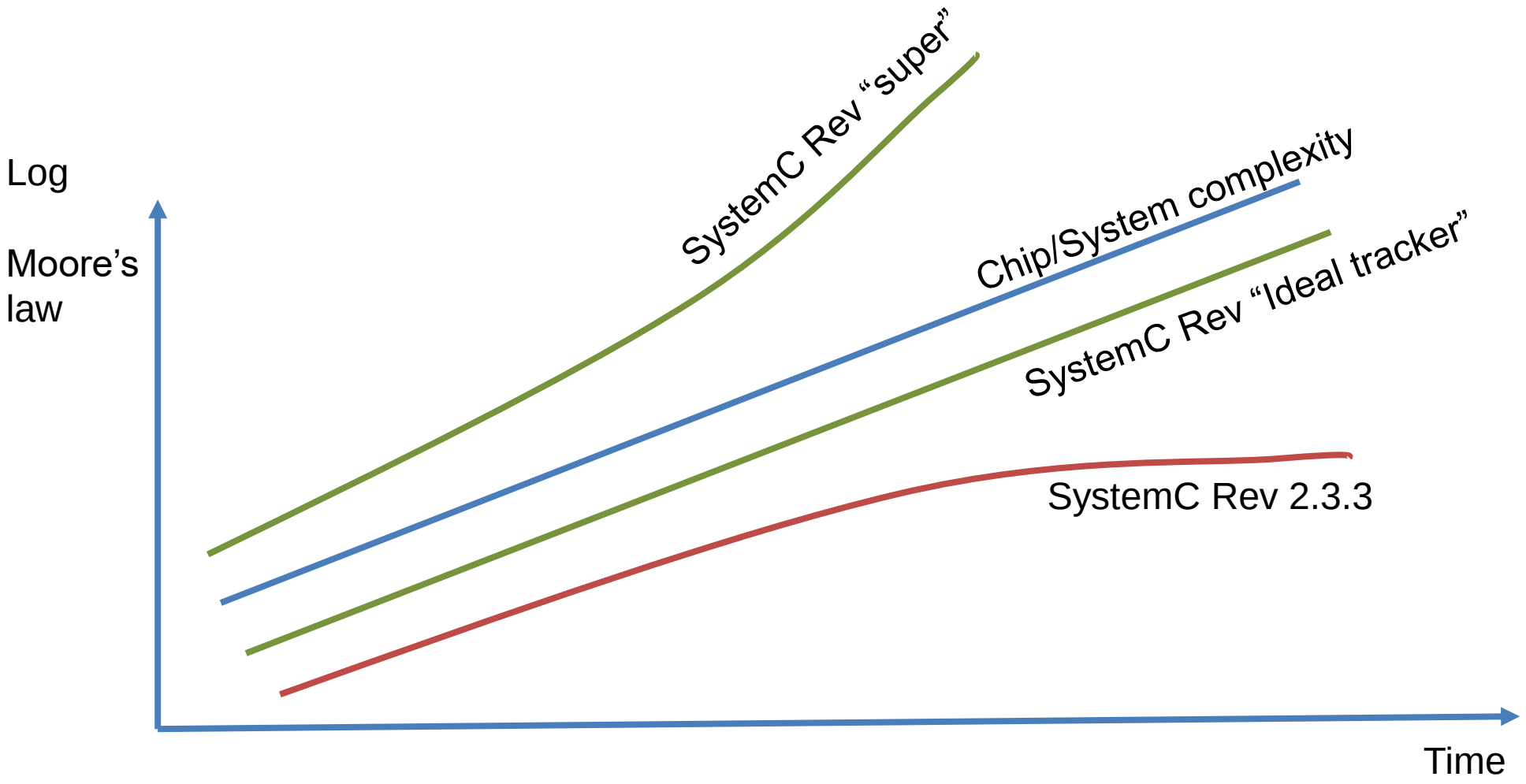
- Most important: community building
  - Putting users of SystemC at the center
  - Establishing a solution hub or playground, a goto for anyone looking for infrastructure, building blocks & models similar to Boost.org
  - Developing libraries of common use: reporting, tracing, instrumentation, configuration etc.
  - Showcasing use and advantages of SystemC as open-source projects and tutorial, videos and alike

# Future of SystemC

- Solutions for parallel simulations as they become imperative with todays designs
- Increase speed of standards development and open it up to the community
  - Outside of Accellera, no-one sees what‘s going on. Suddenly a standard apears for public review.
  - A public playground would allow to spark new ideas and approaches which might go into standardization afterwards

# MANFRED THANNER : NXP

# Is SystemC keeping up with next gen needs ?




Complexity category
- Speed
- Productivity
- Usability

# DAWID STENCEL : APTIV

# SW developer's (App/Middleware) view

- I want fast execution on my **target arch and OS, same toolchain**
- I want to use the model on my PC and my IDE (e.g., Visual Studio Code)
- I do not need to know how the model is built
- I do not want to debug problems in the model
- QEMU is often good enough
- Challenges:
  - Timing issues
  - Missing specialized IP block models (not in QEMU)
  - Pre-silicon – missing models
  - Data injection (e.g., sensor's data)
  - external tool connection (e.g., co-simulation, debuggers)

# Firmware, BSP developer's view

- I want accurate model of specific components and/or entire board
- I want to run unmodified image compiled for target hardware
- I would like to debug and profile it
- I do not want to debug problems in the model
- I want fast execution on my target arch and OS
- Can accept slow execution only if
  - no HW available
  - debug on register level
- External data injection (CAN, Ethernet, SPI, I2C, PCIe, GPIO, analog …)

# Tester and devops

- I have not enough benches to run all tests in 24h
- I need generic hardware model + OS for first fast validation loop (e.g., merge request)
- I need accurate model for component and system integration; can accept slow model for periodic test runs
- I need a way for external data injection and monitoring output
- Always need to minimize costs

# Is SystemC model suitable for SW development ?

Question:

- How to speed up ?
  - multi-threading
  - dynamic changes of accuracy level
  - co-simulation with fast simulators
  - running from snapshot
  - hardware acceleration
  - hybrid simulation with hardware (e.g., NPU, GPU)
- Can SystemC model participate in Federated Simulation ?

# The Future of SystemC

Holger Riethmueller

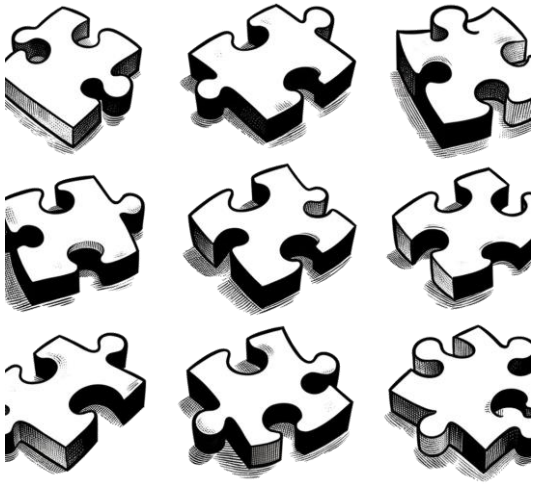
Robert Bosch GmbH

BOSCH

## VP (vECU) Integration

## Simulation Speed


## Chiplets



# Fill y’boots

- Todays Panel: