Verification Futures

The Next Five Years

A selection of challenges from UK and France

22nd November 2012, Munich

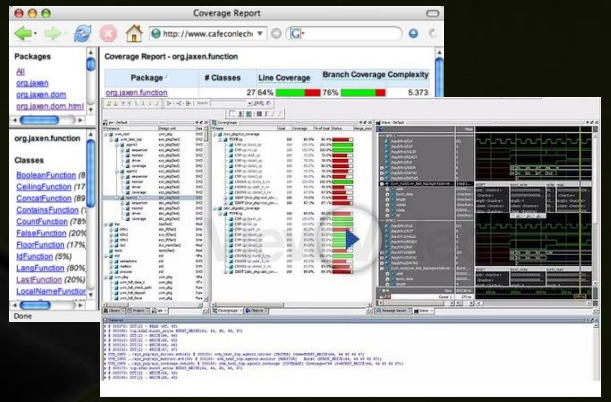
Mike Bartley, TVS

Challenge 3: **Make better use of simulation cycles**

- Random generation covers a lot of the same ground over and over again
  - Verification closure can take too long
- Using directed (e.g. graph-based) generation can make it difficult to hit deep corner cases
  - Bugs can be missed
- Another way?
  - E.g. Random mutation
  - Other machine-learning techniques ...

Performance Verification

- Recently there have been many advances in functional verification
- Performance verification has predominantly been left to individual teams to spin a solution
- Unlike coverage there is no language or EDA support for recording/reporting performance metrics



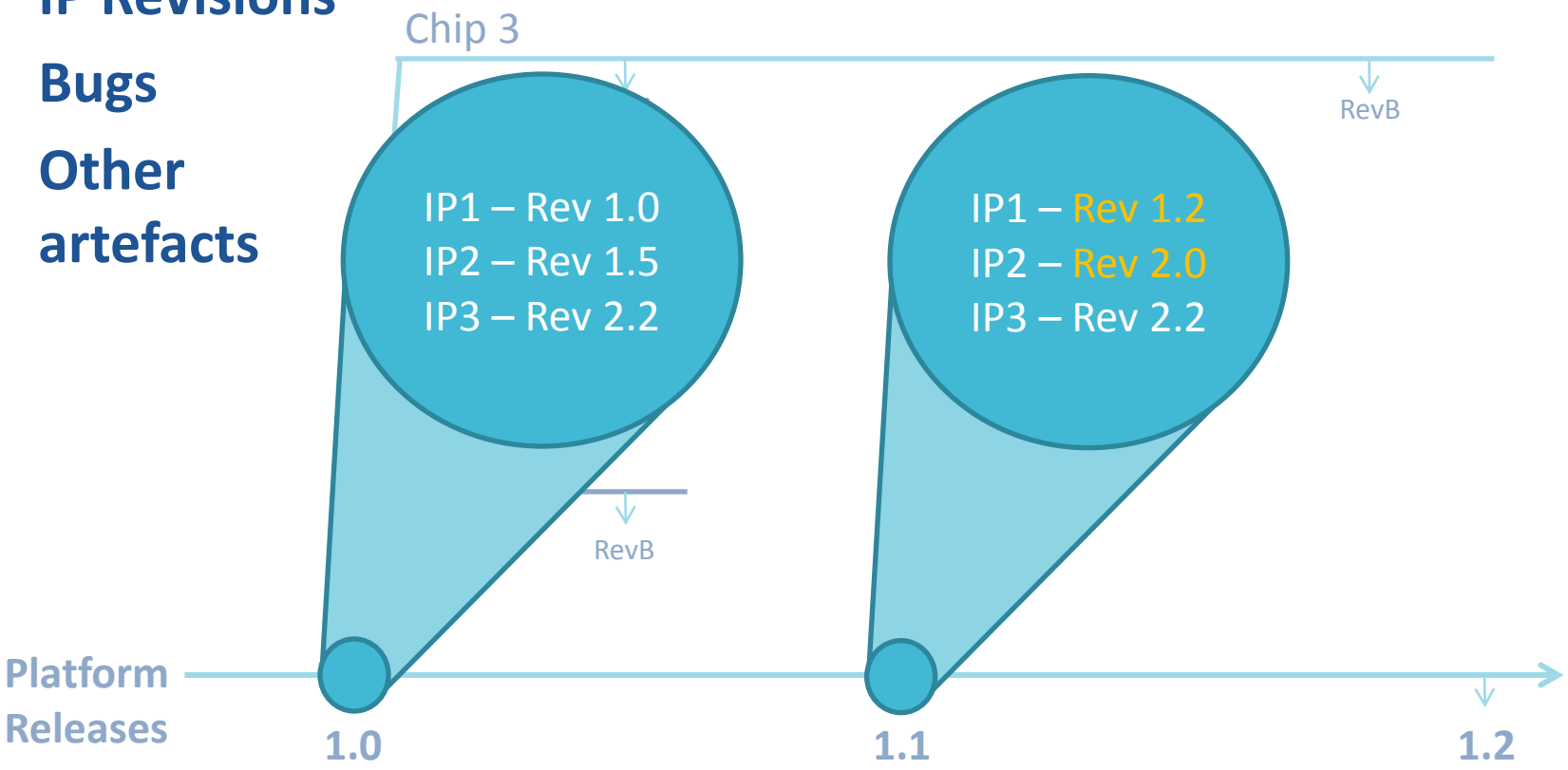
- How do you validate performance of configurable hardware when the use-cases are as flexible as the design?

Where Are We Now?

- Lots of work done on IP core verification.
  - We can get this right so long as we apply enough effort.
- Chips are becoming the problem.
  - The number of cores on a chip means there are real verification challenges at the chip level.
- Verification resources are over-stretched.
  - Need to keep people focused where they really add value and automate other activities.
- Many new flows required.
  - Formal, power, ….

Challenge 1 – Traceability

- Market, customer, product requirements
- Verification status
- IP Revisions
- Bugs
- Other artefacts



3. Mixed Signal Verification

- Functional integration of analogue & digital
  - Late discovery of "simple" problems
- Paradigm shift for engineers
  - Ad-hoc analogue verification technique
  - Digital team consider analogue a "black art"
- Extending MDV to mixed-signal
  - Choice and fidelity of the analogue model:
    - SPICE, AMS, Real number modeling
  - Building a mixed-signal coverage model
  - Choosing appropriate constraints

Challenge #1: Bug Avoidance

- The verification engineers can't take all the responsibilities for providing a correct design
- How to apply more bug avoidance strategies to the development of complex IPs?
- How to design IPs which are fundamentally correct?
- Enabling factors:
  - Engineering cultures
  - Tool capabilities
  - Complexity measurement/benchmarking

Verification Management

- **The challenge to manage huge amount of verification data**
  - Amount of verification data make more complex the risk decision of verification closure
- **Some Directions partially or to be implemented**
  - Refine the verification Metrics
  - Merge the metrics (SOC / IPS – various source)
  - Usage of MySQL data Base
  - Leverage on Business Intelligence tool to support Verification Closure
  - Define metrics non non-functional properties (performance, power, energy, temperature, …)

Leading-edge technology

- How to verify new protocols ?
- Verification before specifications is an issue we have to face
  - “Read ➔ Plan ➔ Execute” is no more reality
- Outsourcing is mandatory
  - Company capacities
  - Company competences
  - Interoperability with customer solutions is mandatory

HW-SW co-verification

- More and more embedded products are HLOS-based (Linux, Android…), most others at least leverage some RTOS
- The final product's SW won't necessarily exercize all of the HW's capabilities…
- …but it might implement some that were not even imagined by the architecture/design!
- So HW-SW co-verification is both a way to narrow-down the ever-increasing verification space…
- …and to increase the <u>useful</u> coverage