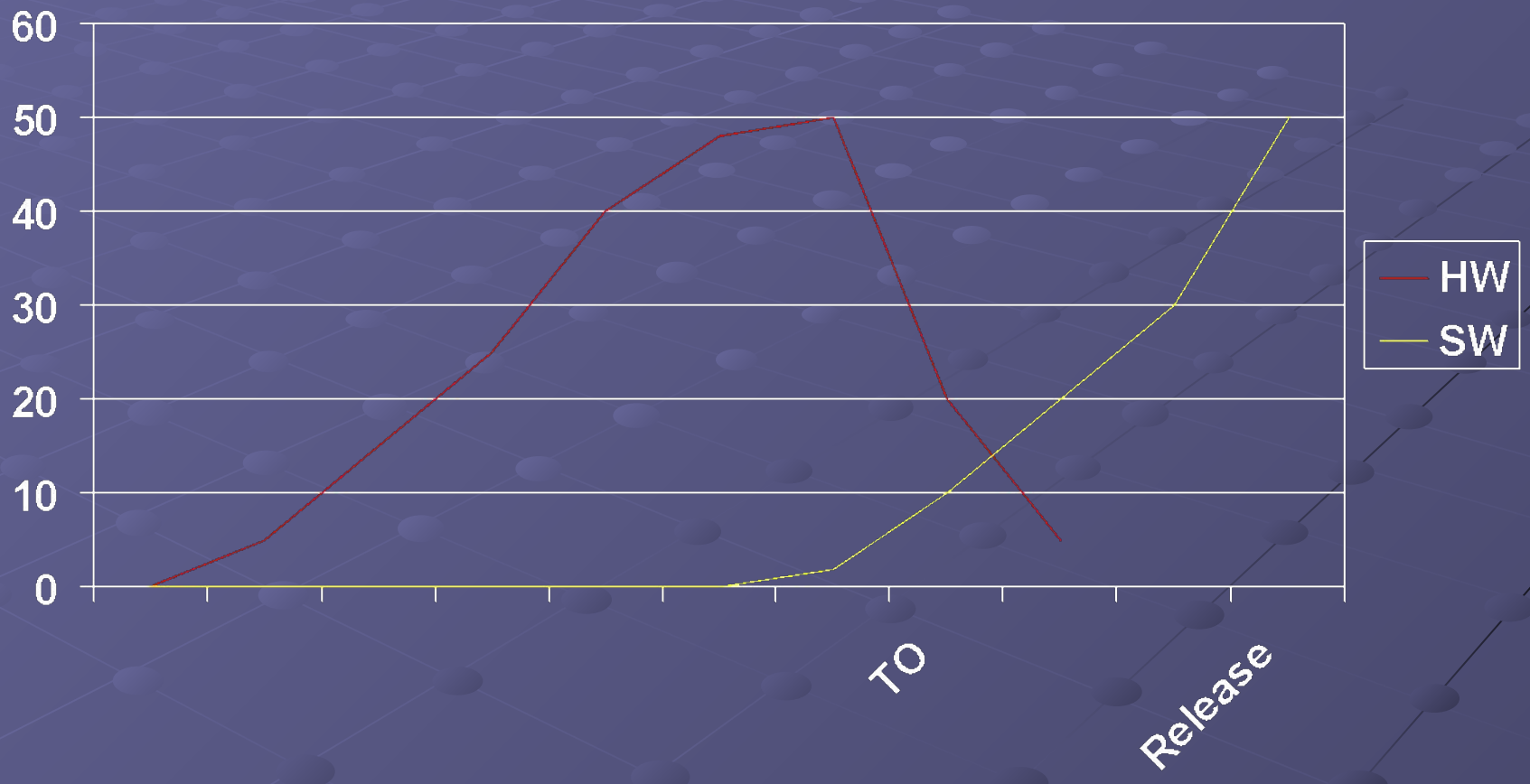
# HW – SW verification

Rony Gutierrez – Mellaox Technologies
Zohar Zafrir – Nokia-Siemens Networks

# Problem Description

- Most of SoCs and ASICs developed today require SW/FW in order to complete functionality.
- SW/FW development usually starts late in the HW development flow.
- Failures on Silicon are harder to debug and verify.
  - Might force a spin / redesign / cancellation.
  - Causes product to arrive late to the customer.

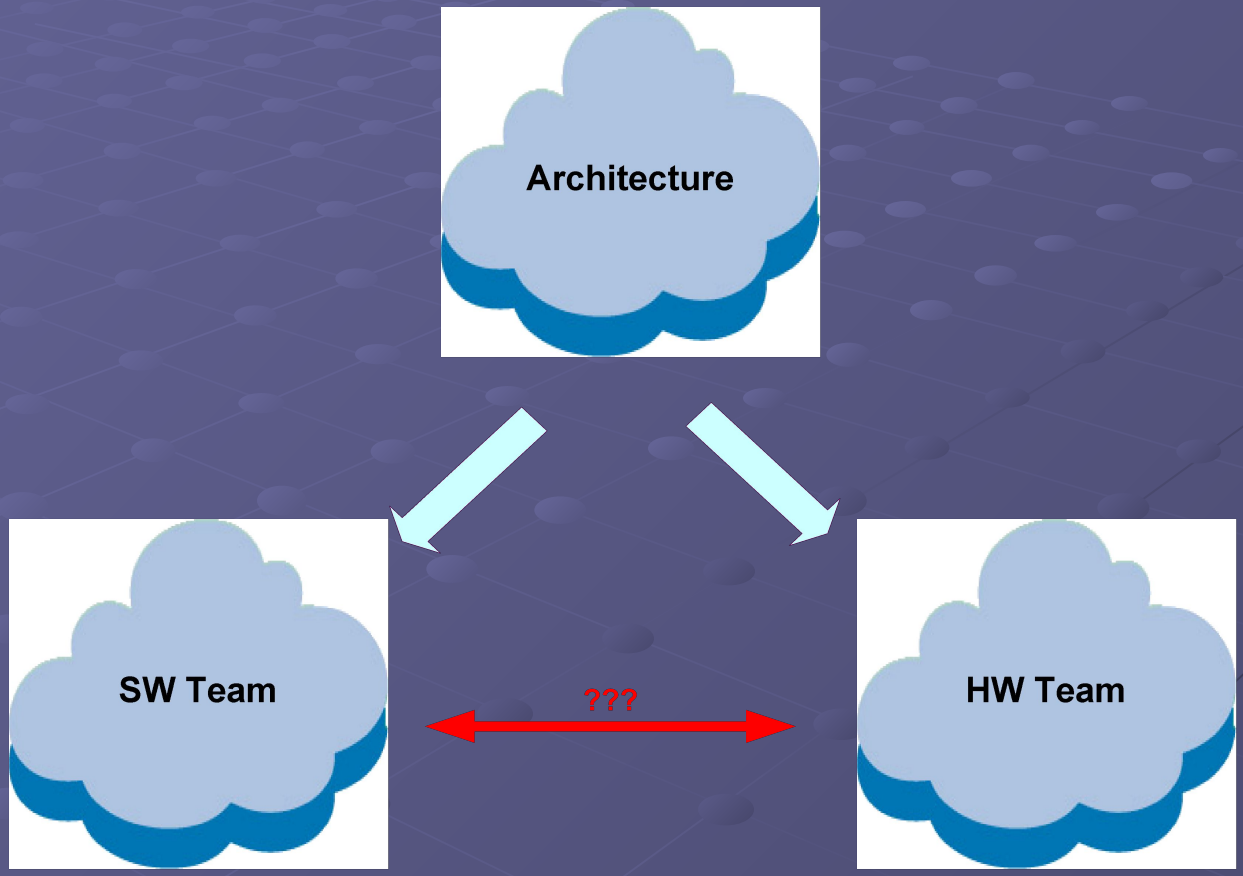
# HW and SW Verification Effort



# What are the Objectives ?

- HW bugs.
- SW/FW bugs.
- Identify Missing features & requirements.
  - Debug hooks.
  - Register / Array Accessibility.
- Debugging / Monitoring tools
  - Allow the HW to “notify” the FW/SW of problems.
  - Interrupts, sticky bits, internal visibility.
- Performance measurement.
  - Impact of SW/FW on performance can be critical if on data path.
- Bring Up Support.
- Reduce Time to Market.

# Where is the real problem?



# Where is the real problem?

- Communication
  - SW guys speak “different” language than HW guys.
    - Serial vs. Parallel design.
    - Abstraction Level.
  - Arch guides both groups separately.
    - Specification changes.
    - Register / Interface changes.
  - How can we bridge the “language” gap?
    - Pseudo-code?
    - Common reviews?
- Is it enough to “imagine” the whole picture?
  - Should we simulate everything to ensure correctness?

# What SW/FW to run?

- Run real FW/SW?
  - Diagnostics
  - Application
- Run “preliminary” SW?
- Run the flows from the verification environment?
  - Exact.
  - Add some randomness.
    - Final SW is unknown.

# What platform to use?

- Simulation?
  - Full visibility, Slow.
  - Not intuitive to SW.
- Acceleration / Emulation?
  - Usually needs different testbench.
  - No HVL verifiers.
- FPGA?
  - Different technology.
  - Less visible than simulation.
  - Time of iterations.
- Virtual platform?
  - Who writes it?
  - Needs maintenance.
- Silicon?

# When should we start?

- HW Design stability.
- SW/FW arch stability.
- Register space access.
- Emulation - ?

# Responsibilities and Staffing

- Who is responsible for the whole system functionality?
  - HW verification? FW ? SW? Arch?
- What is the HW group role during Bring Up?
- How do we pass knowledge between the groups?
  - Reference from verification environment.
  - Reviews.

# Verification pitfalls

- How do we verify performance?
  - Benchmarks?
- How do we verify complex FW flows?
  - On the fly configuration.
  - Power-up/ reset.
  - Tables update.
  - Interrupt routines (long, nested).