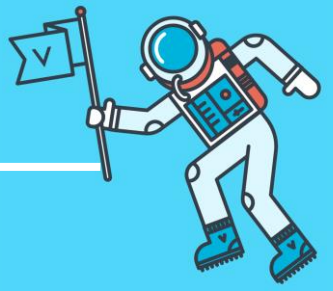
vtool smart verification

The Vtool Friday Tech Club



Various Topics in Full Chip Verification

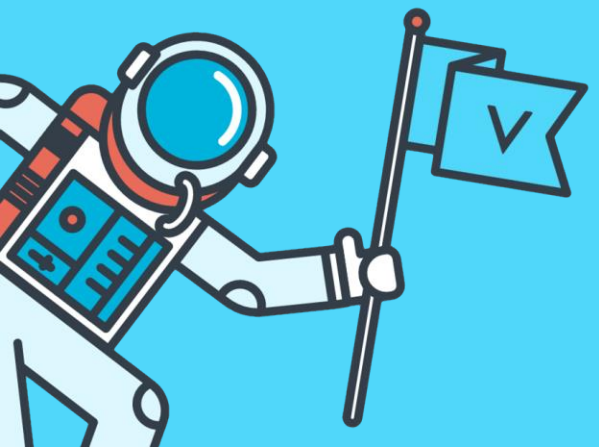
Hagai Arbel

January 24, 2020

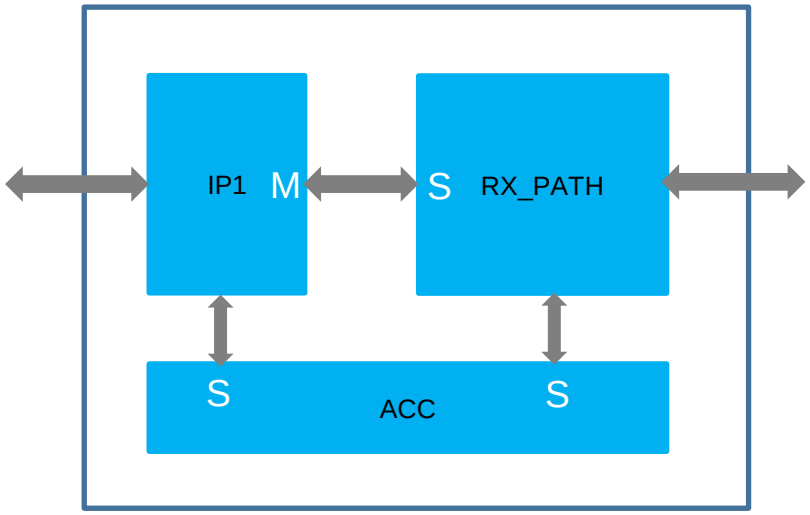
Agenda

- Block level vs. Full Chip
  - Strategy considerations
  - UVM support
  - Coverage on FC and the definition of done
- FC tested features
  - Registers
  - Interrupts
  - IO muxing and connectivity
  - HW-SW co-verification
  - Power, Analog connectivity, Monitor bus

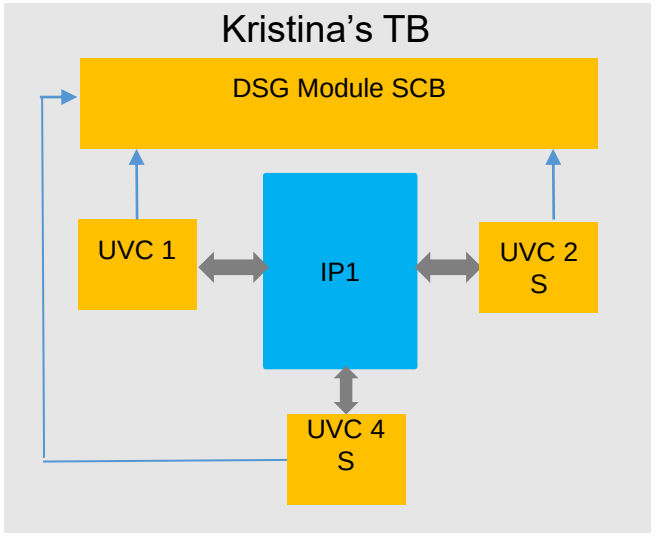
Strategy Considerations

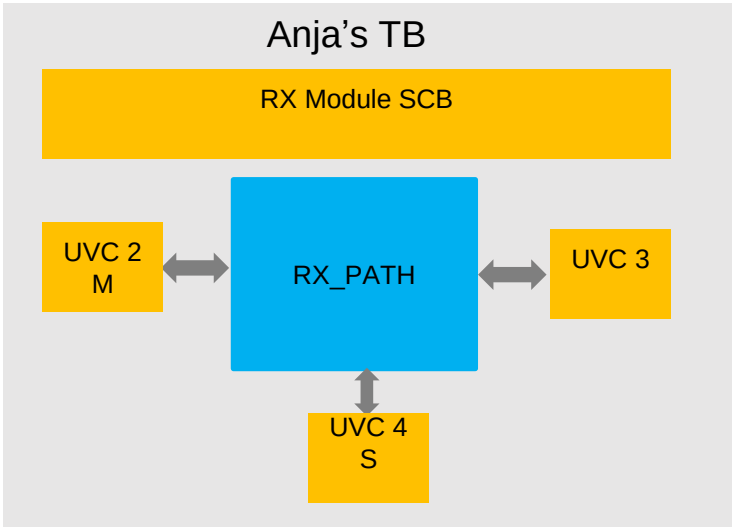


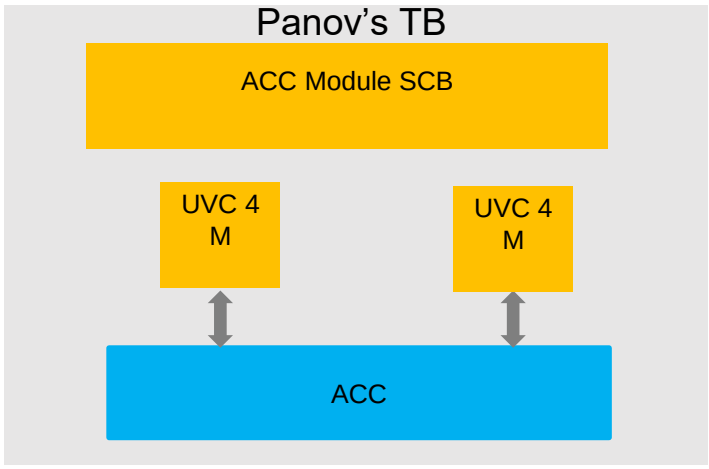
From Module to FC – UVM support



=>

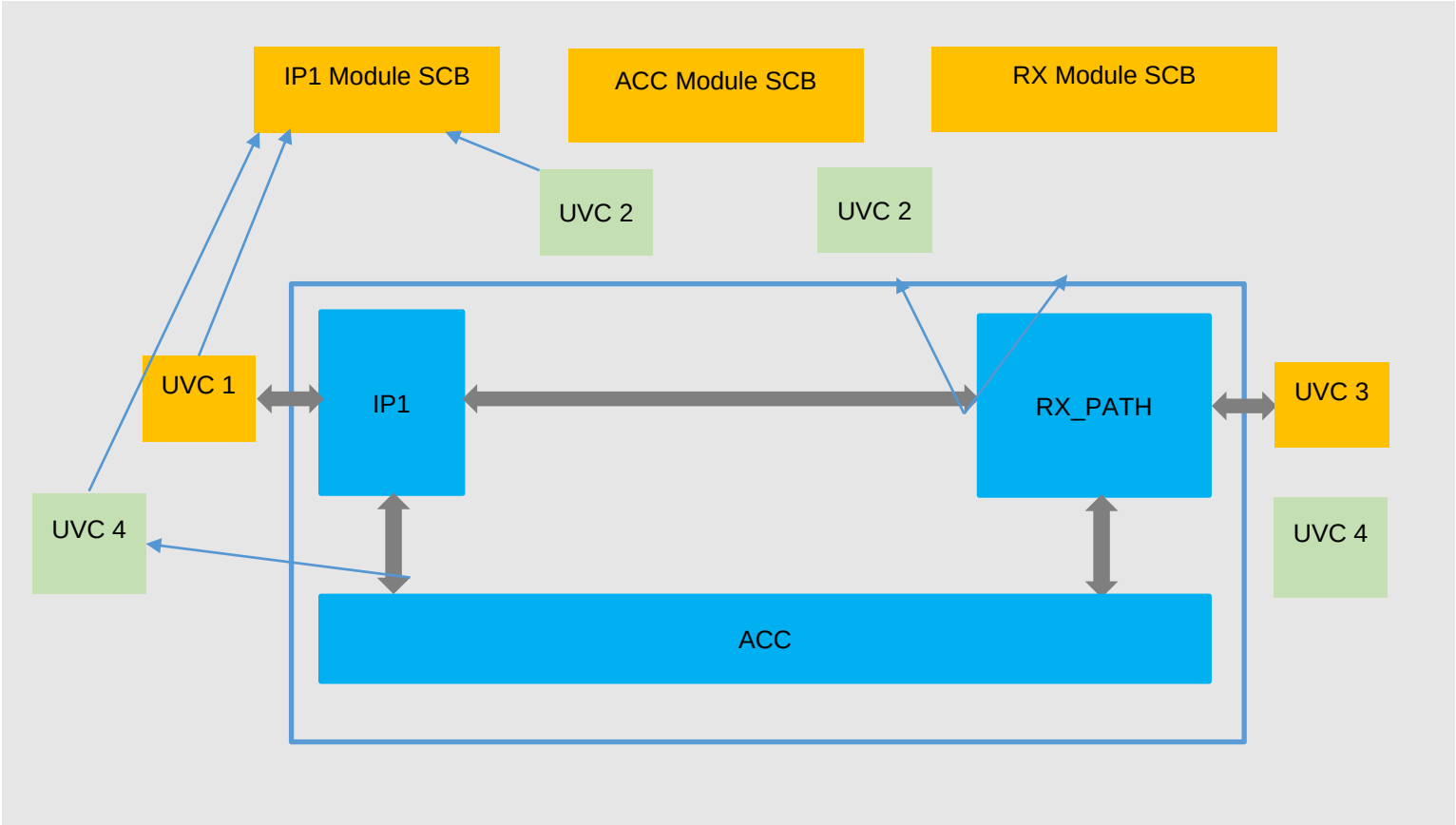






From Module to FC – UVM support

- Modules' TBs are integrated into FC.
- UVC that are now connected to the ASIC's internal I/Fs are turned passive (Monitors only)
- Note that the two instances of UVC 2 must be connected to the same hierarchy. (Why?)
- UVCs continue to feed the modules SCBs and the SCBs continue to check.
- Sometimes, additional End-t—End SCB is created.
- Now, this is all by the book – All the 'buts' are in the next slide.



How we decide to which blocks the chip is divided

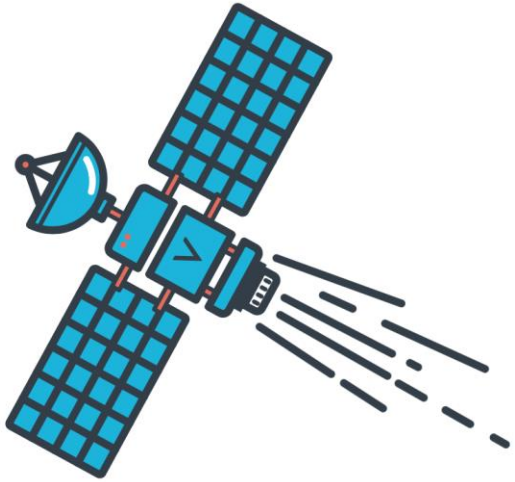
Topic	Verify the feature in Block Level	Verify the feature in FC
Simulation time	Short	Long
Controllabnility and Observability	Better – high coverage	Worse – Low coverage
Closer to real scenario	Depends on sequences, can be wrong	Closer to real scenarion by definition
Effort	Higher	Lower
Work in parallel – Team size	Can work in parallel, even if RTL of some blocks is not ready	Need the entire RTL, harder to divide
Design type – SoC vs. Data chain		It is easier to divide the work in FC for SoC
RTL maturity	Do we need block level at all? What if this is a verified IP?	

What before How

We first must know WHAT we want to verify.

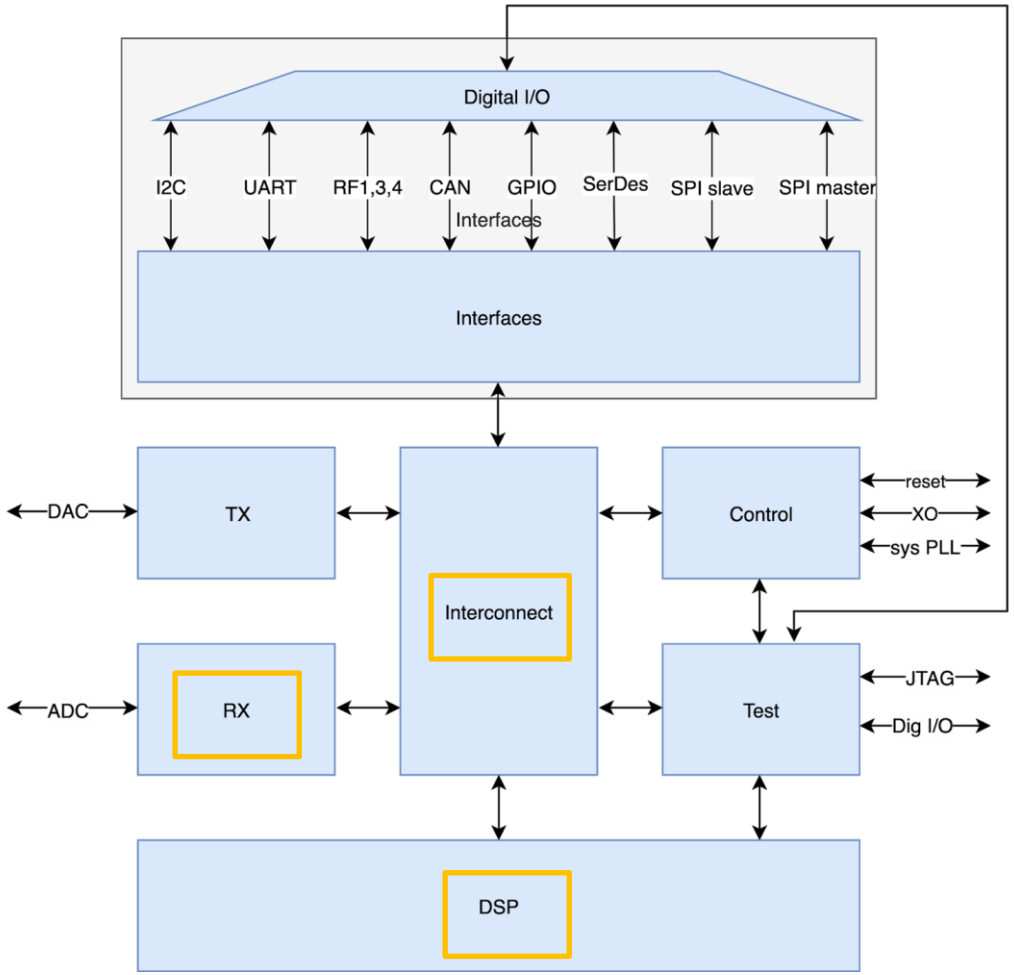
Only then, we can decide HOW we'll verify it.

Let's see how this applies in Block-FC decisions.



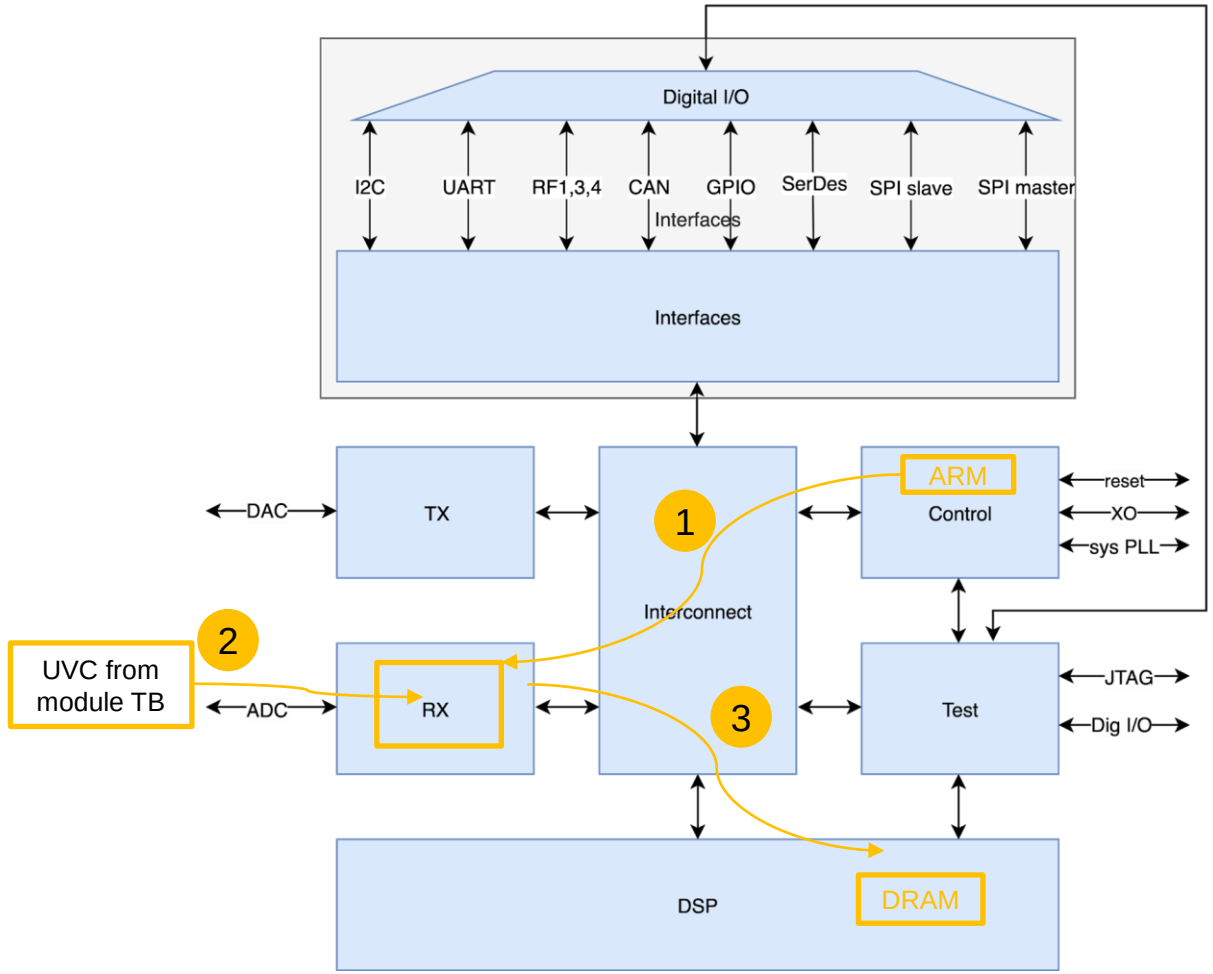
A few examples

- The **RX** has a module VE and integrated to FC.
- The **Interconnect (NoC)** is an IP – No need for a block level. We stress it on FC because:
  - We need to check it was configured properly (RTL is generated by CFG)
  - We need to check it performs within our ASIC architecture.
- The **DSP** is an IP but it contains some glue logic we touched. So we must cover this functionality in FC.

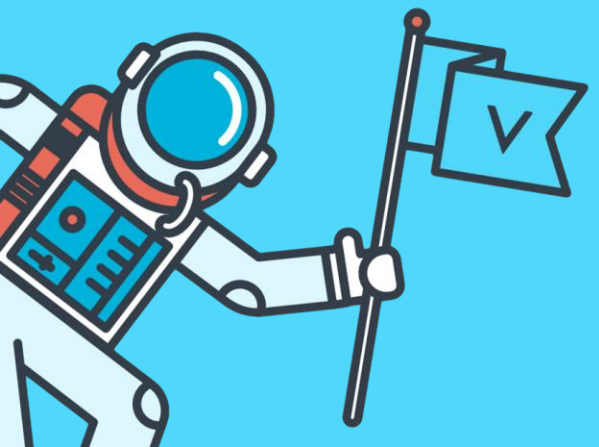


Example: RX FC testing

- When running RX tests in FC, we check, for example:
  - That the RX I/Fs to the NoC are working properly – we did that with UVC in module level – now it is the real NoC RTL.
  - The ARM CFG the RX registers – we check proper connectivity and address decoding from the ARM to the RX module.
  - We check proper connectivity from the RX module the the DRAM, that is located within the DSP subsystem.

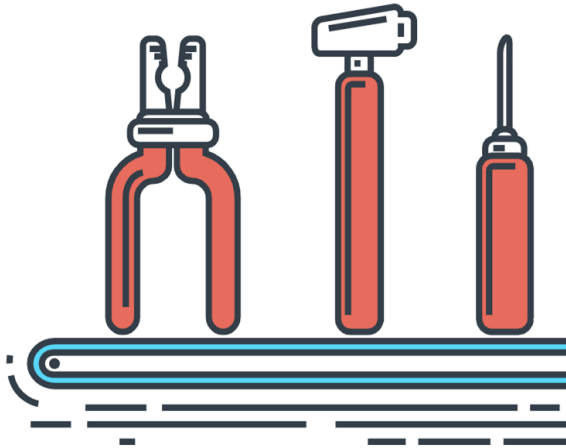


FC tested features



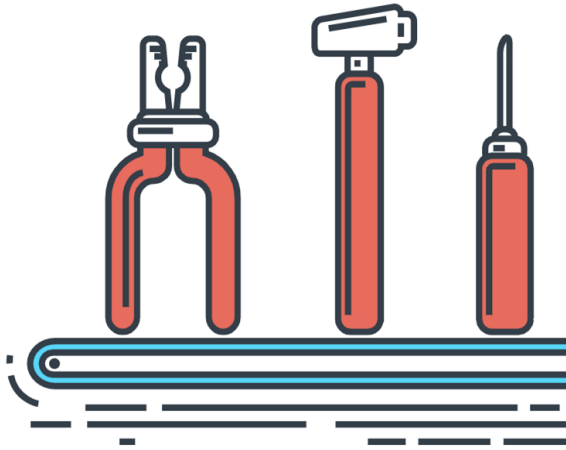
Registers

- We want to check that each master (ARM, DSP, External Host, etc.), can access all the registers that it should.
- Do we need to check every register? Do we need random?
  - If we do not have module level register test for a certain module – Then probably yes.
  - If we do have full module level register test, we still need to verify:
    - Proper address decoding to the block.
    - Toggle of all addr/data bits.
    - Concurrent access from different masters (?)

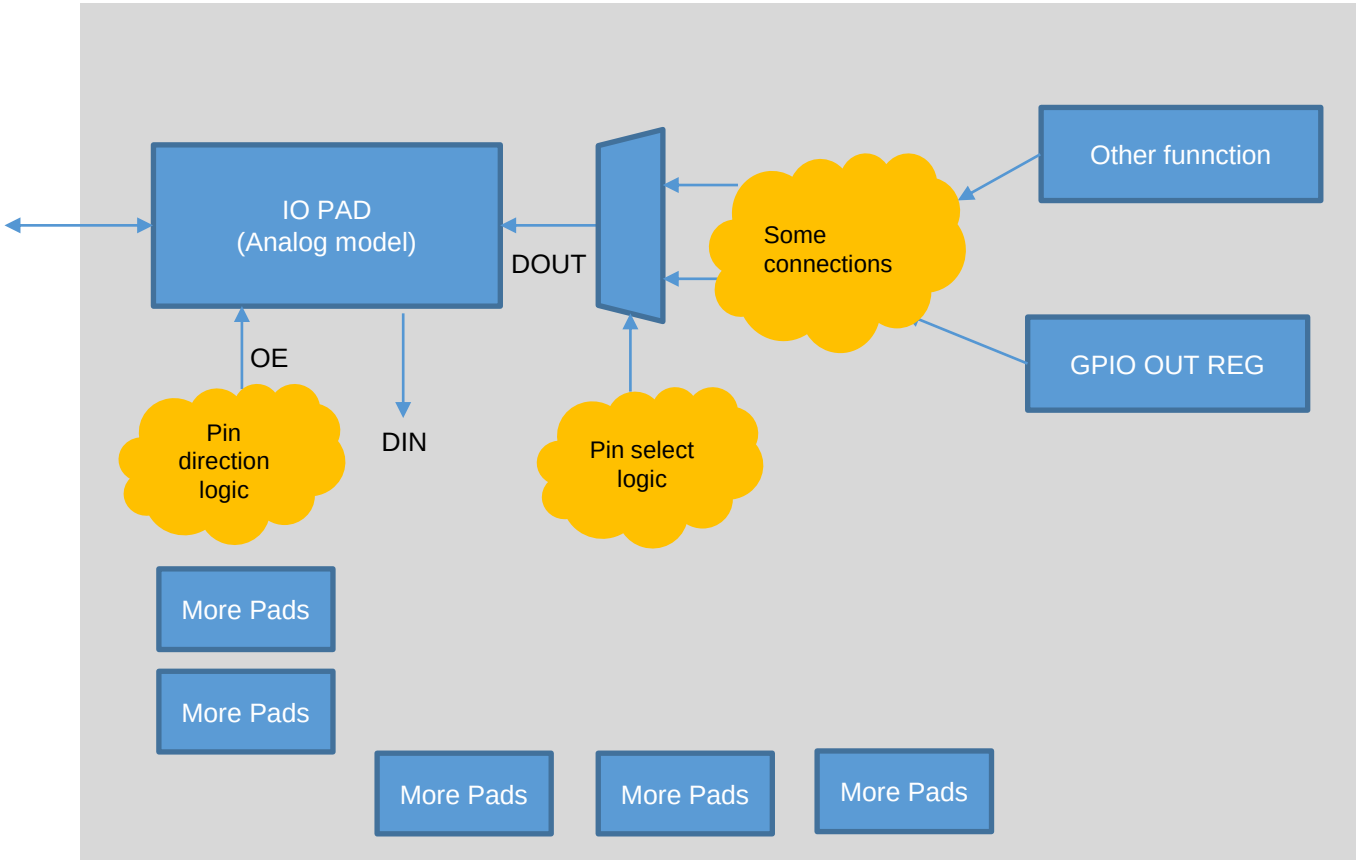


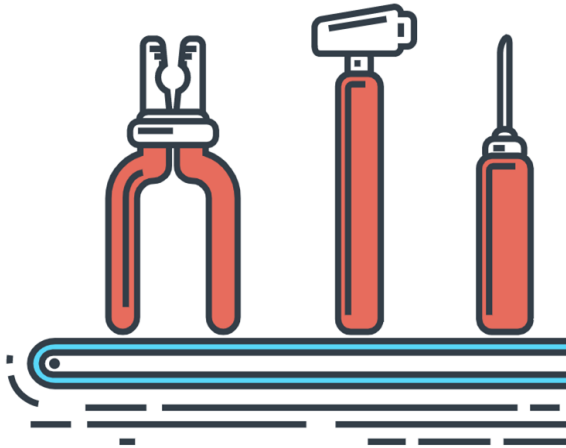
Interrupts

- We want to check that each interrupt is propagated to the CPU (or CPUs).
- Do we need to create functional scenarios to generate all interrupts?
  - If we do not have module level interrupt test for a certain module – Then probably yes.
  - If we do have full module level interrupt test, we still need to verify connectivity. Sometimes forcing the source and checking at the CPU input is enough.



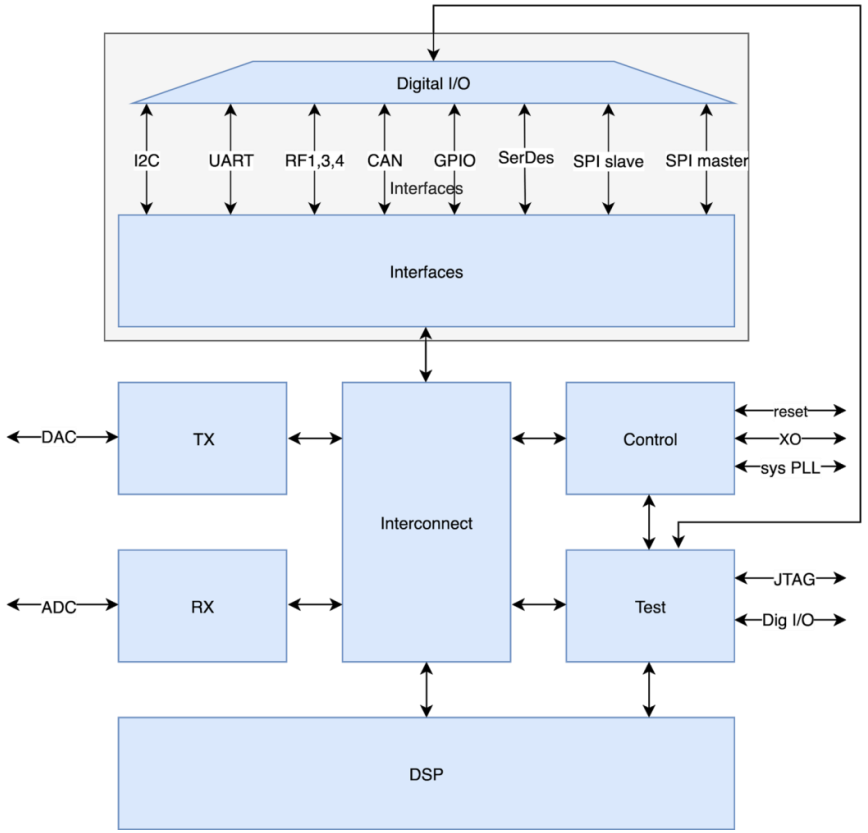
IO muxing and connectivity





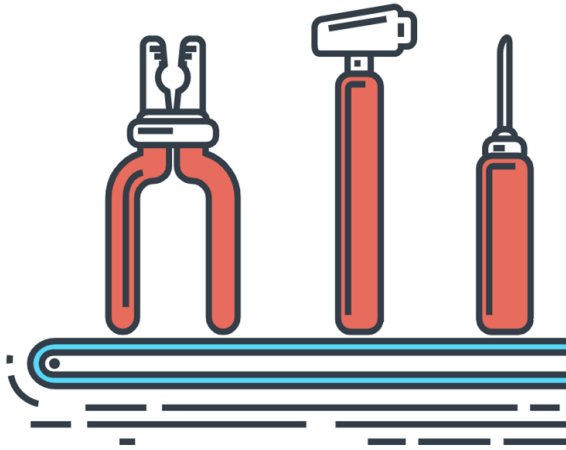
HW-SW Co-Verification

- We typically do not have to test the CPU itself – It is a proven IP.
- But we do need to test the connectivity and that the system performs as a whole.
- We can write C code for the CPU, compile it to binary, load it to the code memory @simulation start and run the test.
- It needs synchronization and inter-operation with the UVM TB.
- We can also connect a VIP at the CPU bus output (AHB, AXI) and have a VIP performs the “CPU’s job”
- This is a subject for a whole new lecture. Typically, we use a combination of both.



Power, Analog connectivity, Monitor bus

- Power domain are typically tested in FC – Either only connectivity or as a full power-aware simulation.
- The Analog parts within the FC are represented by models.
  - We either test only connectivity; or
  - We run AMS – Analog Mixed Signal simulations.
- In almost every ASIC we will have some kind of Monitor Bus
  - Muxing of internal ASIC key signals onto the ASIC output pins.
  - Used for on-board debug (Real LAB Silicon debug)



Questions?
Thank you!