Physical synthesis/layout issues specific to FPGA-based emulation

International Symposium on Physical Design’2026

Helena Krupnova, PhD

March 16th, 2026

Outline

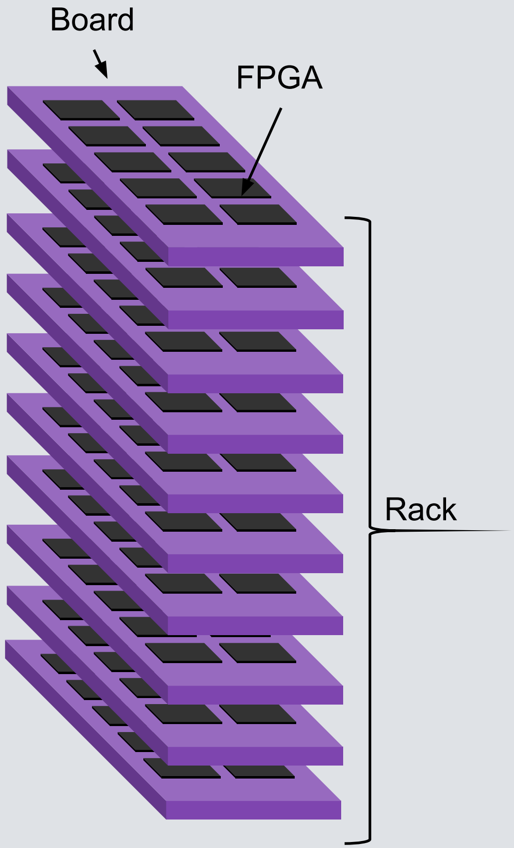
- Introduction to Synopsys Emulation Solutions – ZeBu
- ZeBu Compiler – FPGA implementation flow
- Challenges to compile large customer designs to FPGAs
- FPGA compilation - in depth
- Optimizations and research directions
- Conclusions

Synopsys HAV Product Family

ZeBu Emulation Platforms

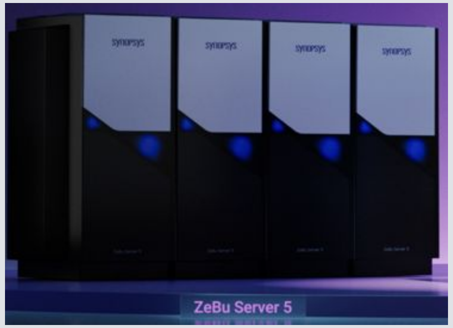
- Based on AMD Xilinx FPGAs
  - AMD Versal™ Premium VP1902
- Emulation-dedicated HW (ZeBu)
- Prototyping-dedicated HW (HAPS)
  - Possible mixed usage (EP-ready)
- New generation every 4-5 years
- Various types of FPGA architectures:
  - Fixed interconnect
  - Flexible interconnect through cables
  - HUB-based interconnect (reconfigurable crossbars)
  - HPIO (single-ended and differential) and MGT (transceiver-based)
  - 4/6/8/12 FPGAs per board

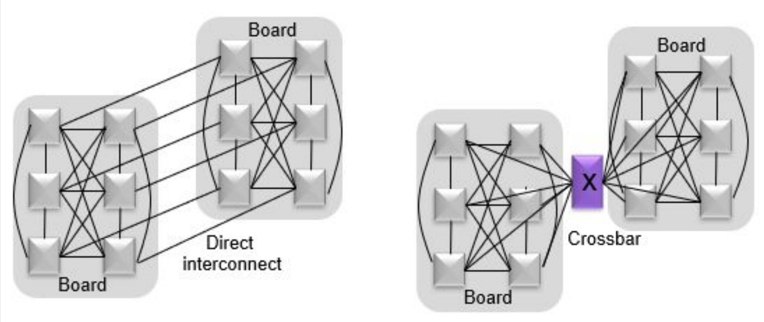
⇒ taking advantage of best-in-class HW technologies for scalability and performance

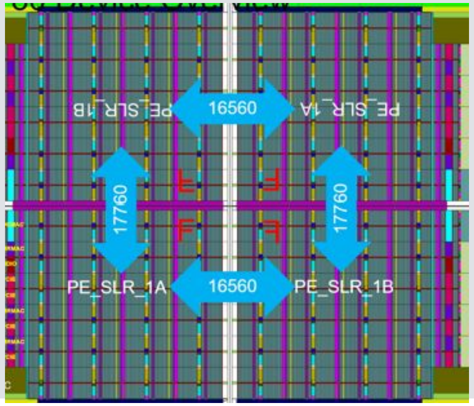


FPGA-Based Emulation Systems – Physical Design Perspective

- Macro-architecture
  - Physical aspects: number of racks, footprint
  - Number of FPGAs per board, number of boards per rack
  - HUB connectivity
  - Cooling
- Board architecture
  - Inter-FPGA interconnect choices – HPIO, transceivers
  - Memory ressources
  - Debug ressources
- Micro-architecture
  - Implementation within FPGA
  - FPGA architecture aspects, die organization
  - Firmware implementation

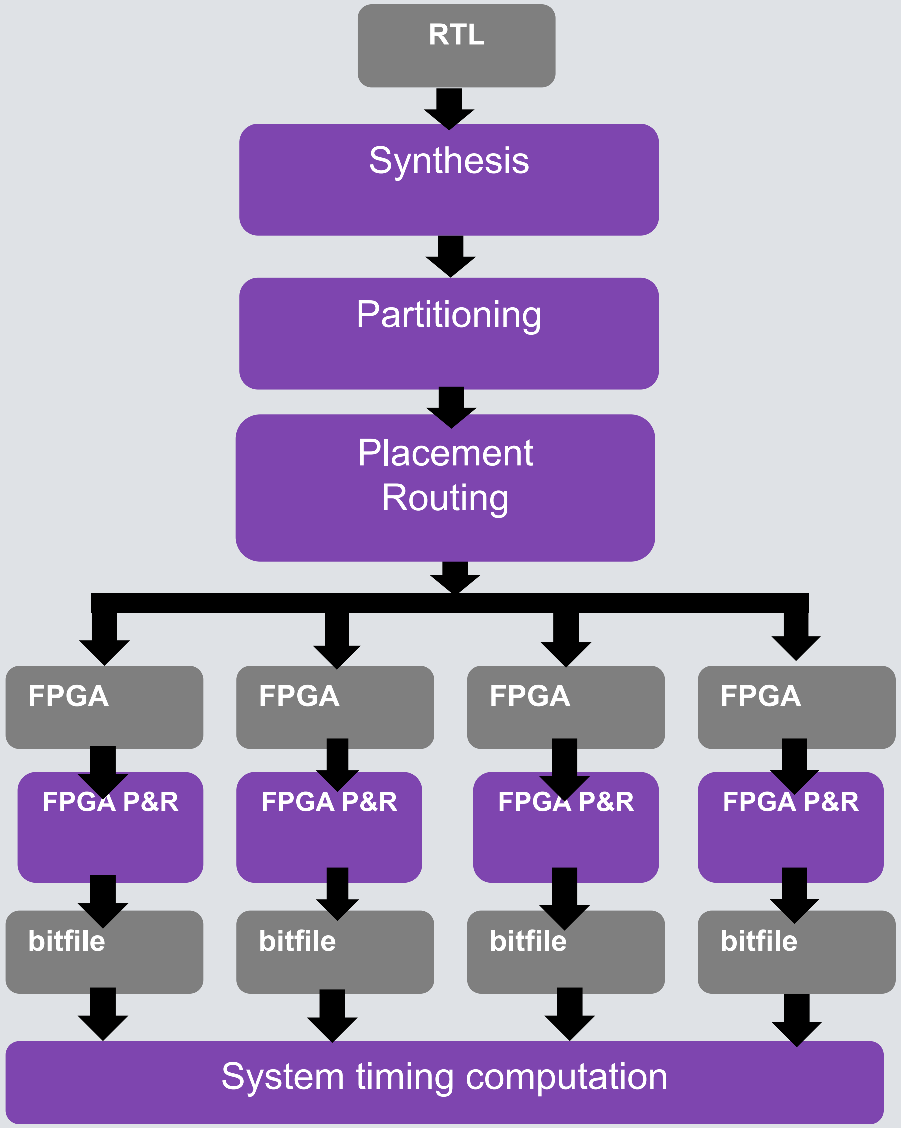






The emulation Compiler – ZeBu implementation Flow

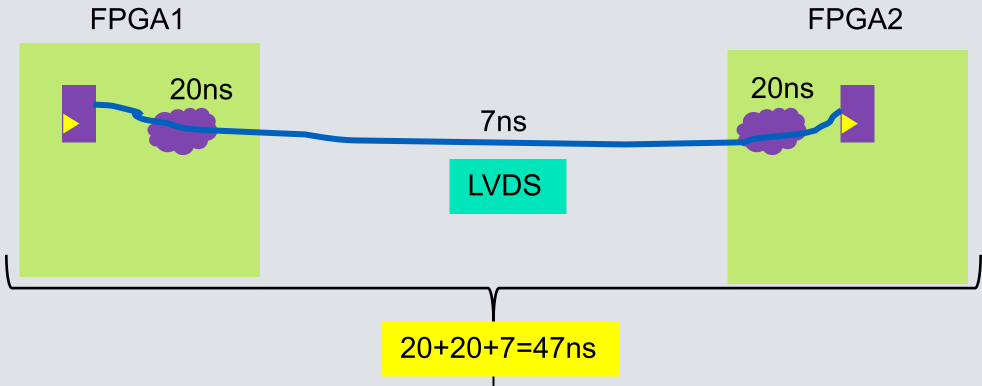
- Leverage parallelism and multi-threading to scale
- Compiler is adaptable for different types of HW architectures
- Multi-level compile flow – system/board/FPGA/++
- FPGA compilation is the biggest contributor to compile time
- Compile success == 100% of FPGAs produced bit files
  - 1 FPGA fails no emulation model to load into the HW
  - Decisions taken by tools on top of the flow may help compile success or lead to failure
  - Stable and reliable compile is MUST
- Compile flow can be setup for:
  - quick compile
  - performance
  - debug



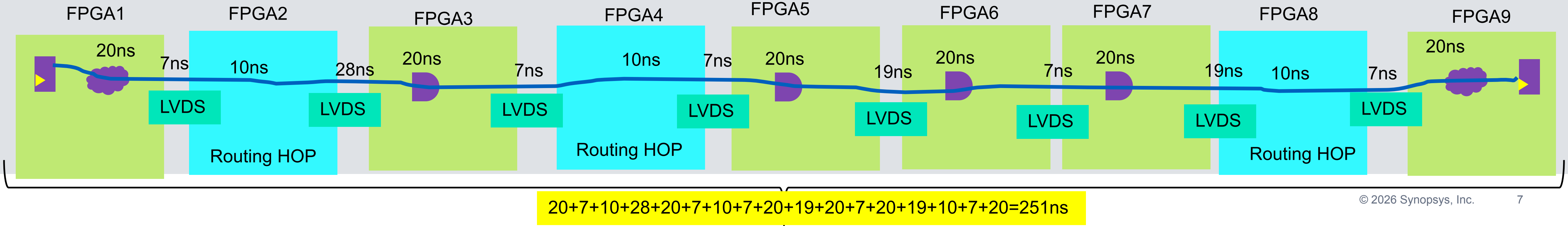
Emulation Model Performance

- ZeBu is the fastest emulator as of today – emulation model performance can cross 10MHz
  - ZeBu-EP and ZeBu-200 have highest performance potential due to customizable architectures
- Emulation frequency is computed based on adding:
  - Internal FPGA delays
  - Interconnect delays
- Optimization focus:
  - Number of FPGAs with logic
  - Number of routing HOPs
  - Interconnect delay
  - Number of logic levels

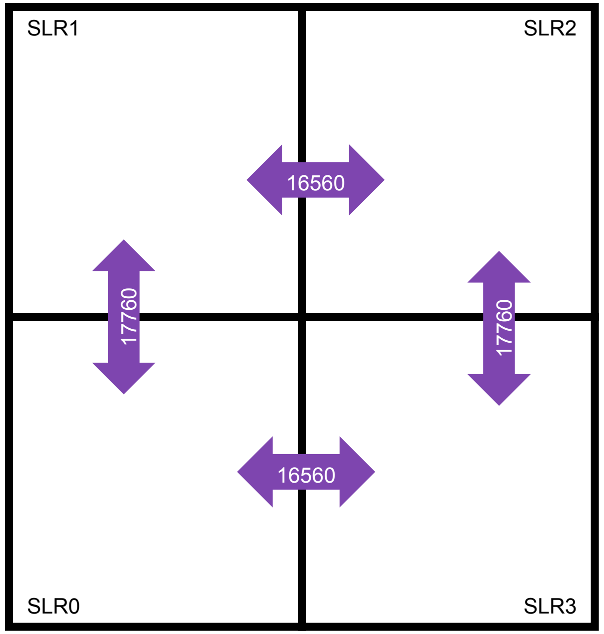
Optimized Emulation Model



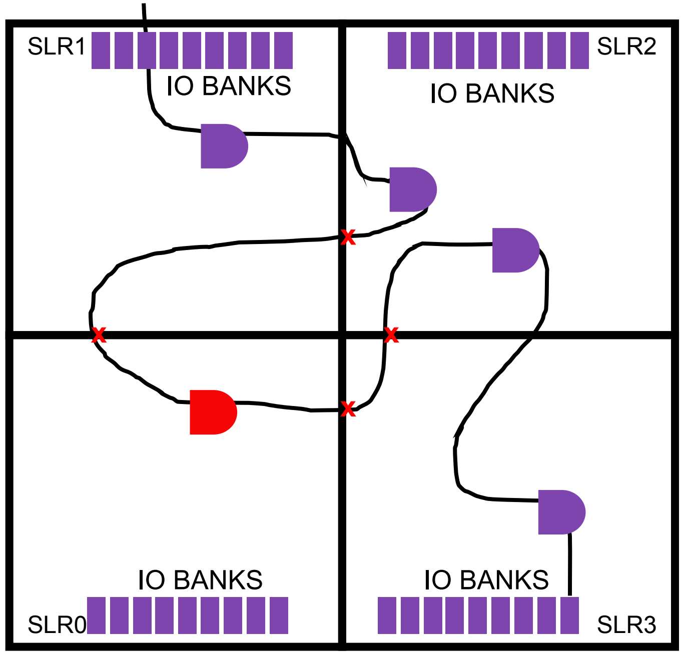
Not Optimized Emulation Model



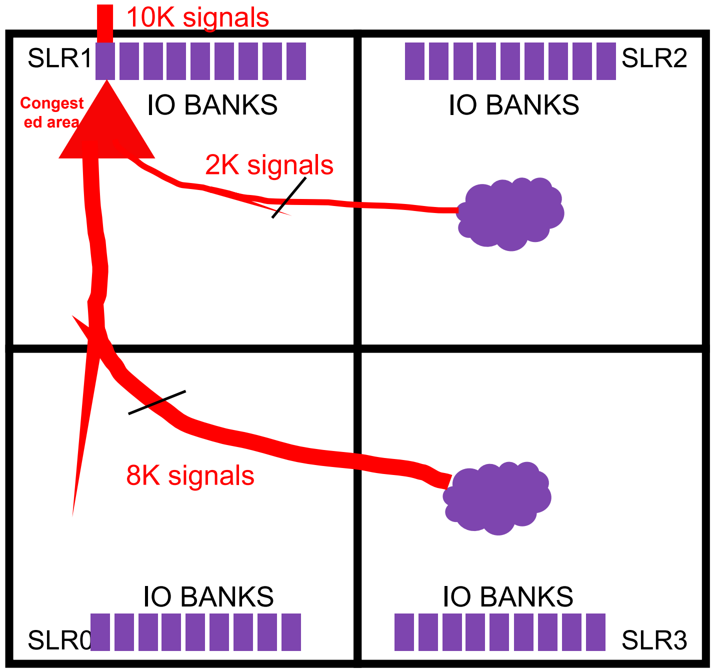
The FPGA Compilation – AMD VP1902



- 4-die chip
- Square (no diagonal interconnect)
- Limited number of SLLs (Super Long Lines)
- IOs on top and bottom

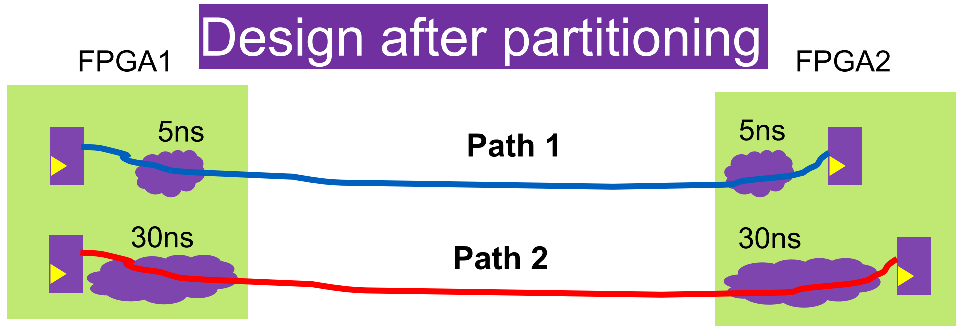


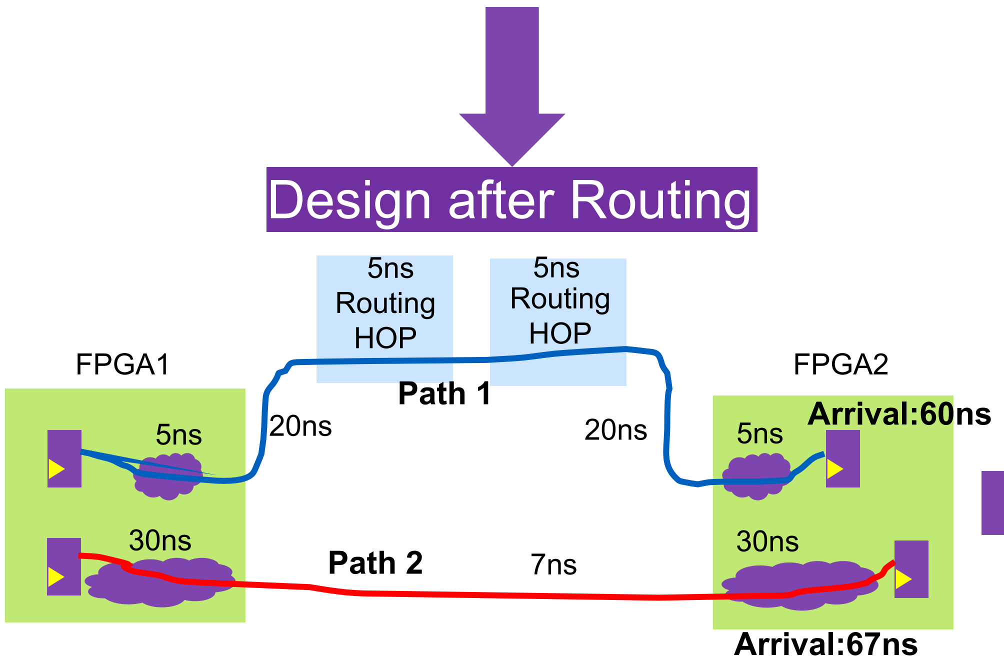
- Die-level partitioning is important for performance as well as compile-ability
- Internal FPGA delays can dominate over platform delays when we target ~10MHz performance



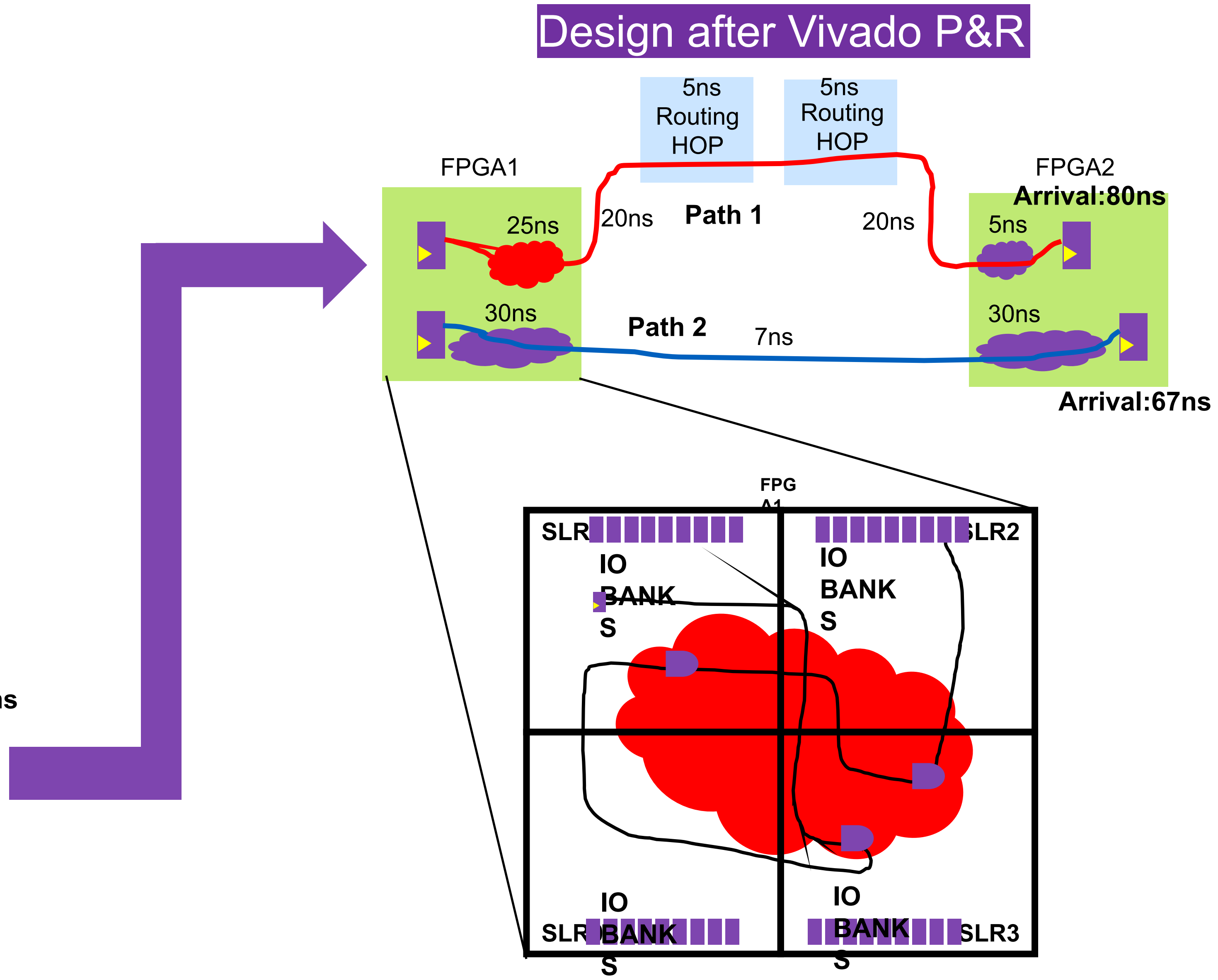
- High interconnect due to dense customer designs can bring large number of signals to the same IO bank
- Congestion around sockets is a very high risk
- SLL saturation is one of the major compile-ability risks

Internal Delay Prediction

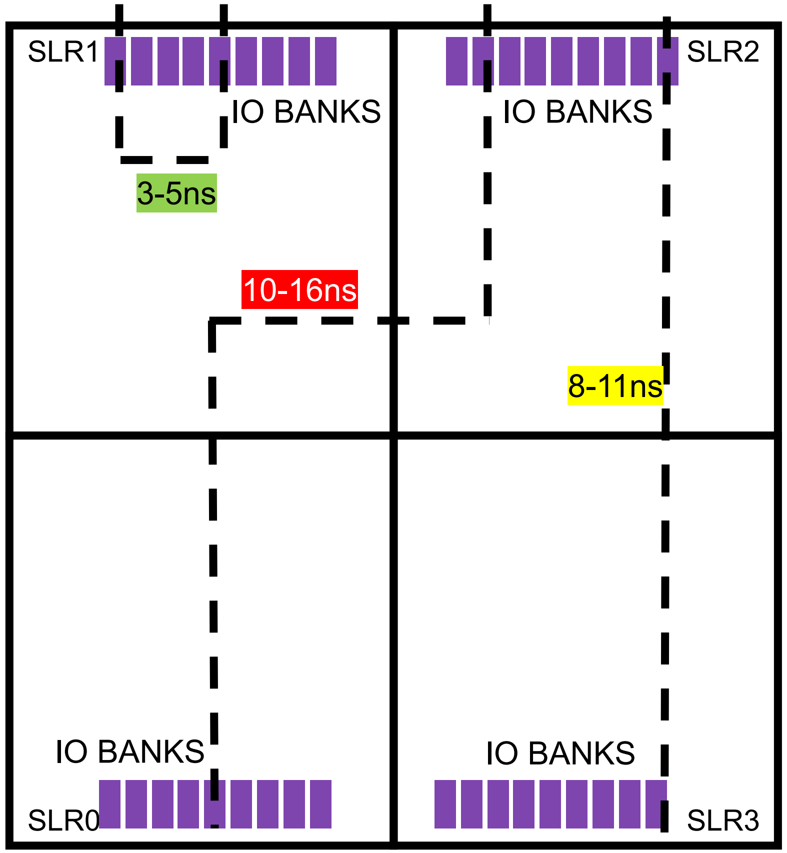




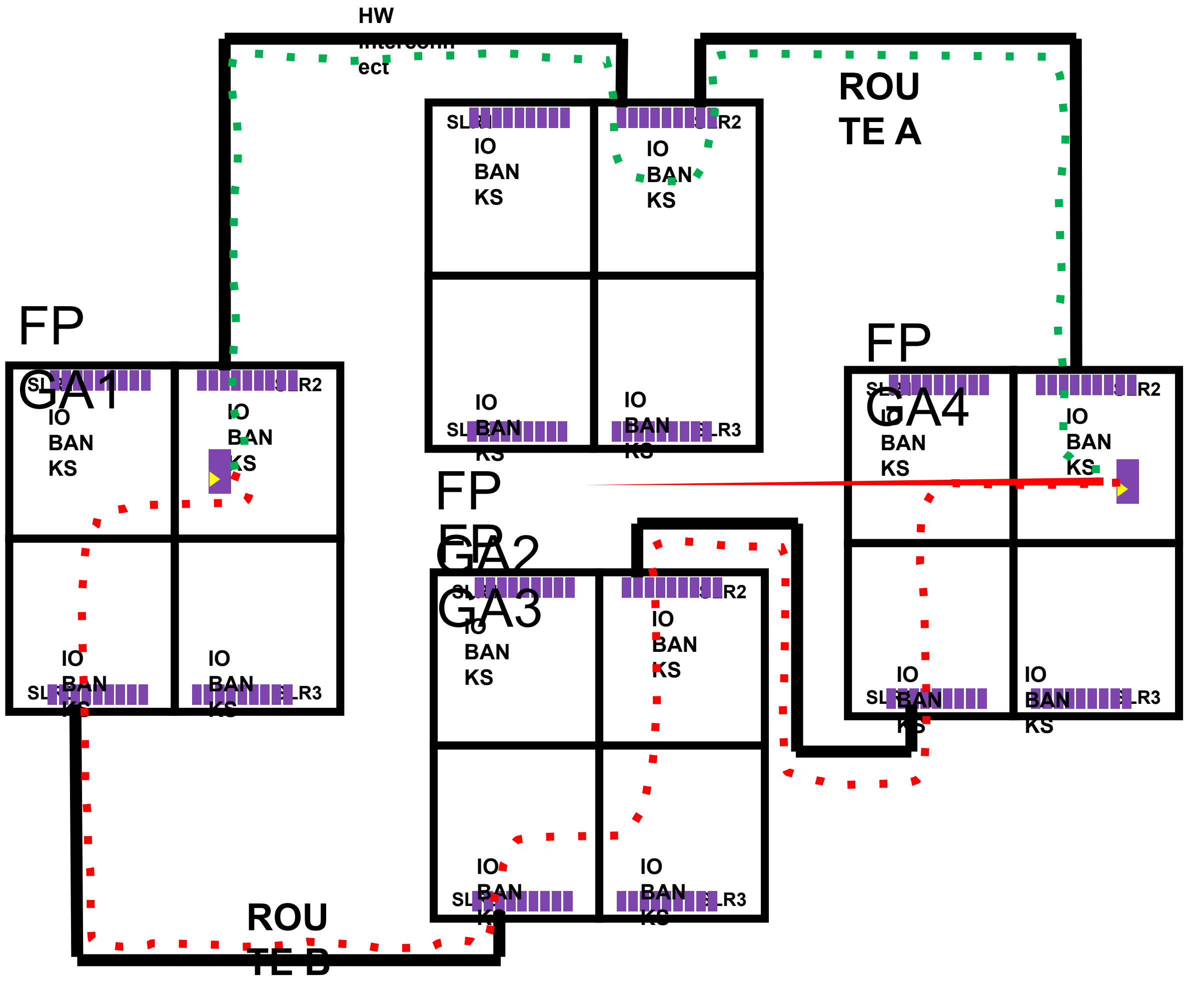
Routing is using predicted delays to take decisions



Congestion-driven Routing



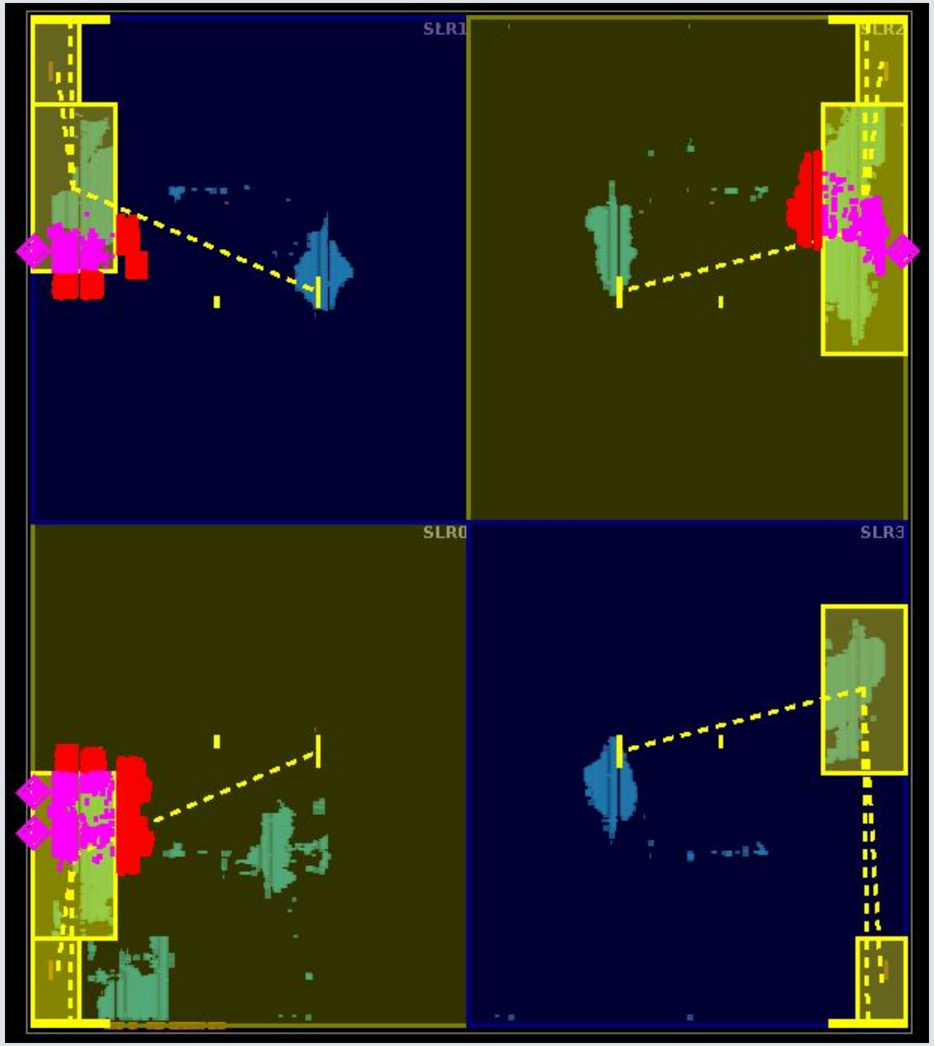
Routing HOP will have impact on compile-ability and performance



ROUTE A – optimized
ROUTE B – sub-optimal

Internal FPGA Floor-Planning

- Inter-FPGA, inter-module connectivity may impact FPGA routability
- Mixing high-speed logic (memory controllers, MGTs, XPIO sockets) with DUT logic is creating heavy pressure on Vivado P&R tool
- Guidance for Vivado P&R is key to achieve stable compile results
- Internal FPGA implementation is architecture when building the entire solution
- Tuning and customization represent significant investment



Conclusions

- Several hundreds of FPGAs need to be successfully compiled to obtain an emulation model
- Numerous challenges to be solved by emulation compiler – design synthesis, partitioning, placement, routing, timing analysis, FPGA place & route
- Top-down and bottom-up dependencies between different implementation layers:
  - top-down decisions may impact routability,
  - FPGA device topology requires smart decision on placement and multi-FPGA routing, and even board and system design
- New generation platforms □ new level of complexity: internal FPGA architecture
- Building an emulator == multitude of complex problems to solve!

Thank you