Hardware Verification for High-Performance Designs in the Next Generation: Towards More Scalable and AI-Driven Techniques

Arun Joseph, STSM, IBM Electronic Design Automation, ISDL
Chakrapani Rayadurgam, Cache Verification Architect, IBM Hardware Verification, ISDL

DESIGN AND VERIFICATION
DVCON
CONFERENCE AND EXHIBITION
INDIA

Author Bio

Author Bio

Outline

- High performance design context
- IBM methodology for verification
- AI/ML techniques
- Collaborative debug platform

Concept to Product

Growing size and complexity

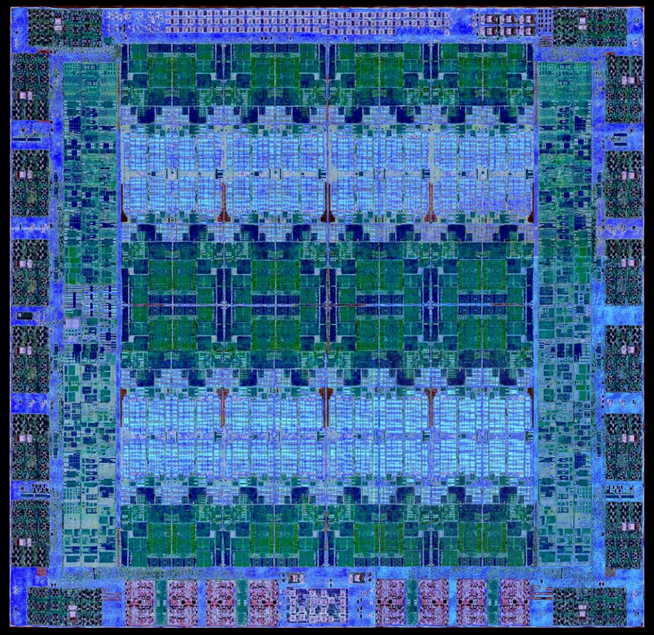


Figure 1 | Reference: ISSCC'22

IBM Power10 is the most secure and reliable server platform in its class

Not just H/W. WE do HW/FW/SW to full systems

7nm bulk technology | 602mm ^{2} chip | 18B transistors

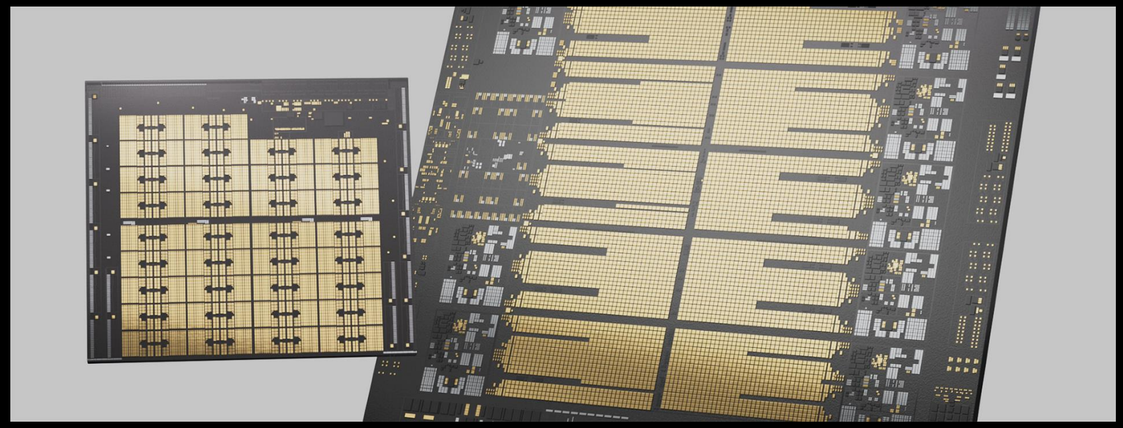


Figure 2 | Reference: Telum

Telum2 launched in HotChips'24

IBM Telum is our first processor designed with on-chip accelerator for real-time AI inference

5HPP | 43B transistors | 600 sq mm *

* Projections from preview hardware

Outline

- High performance design context
- IBM methodology for verification
- AI/ML techniques
- IBM debug platform

AOD | Aspect Oriented Hardware Design | Motivation

- ✓ By separating the aspects, the POWER10 simulation showed a ~24% increase in cycles-per-second on the RTL models over P-RTL. Similarly, model compilation time reduced by ~35%.
- ✓ Overall methodology helped lab bring-up of POWER10 wafer in ~1/8th time, compared to prior generations.

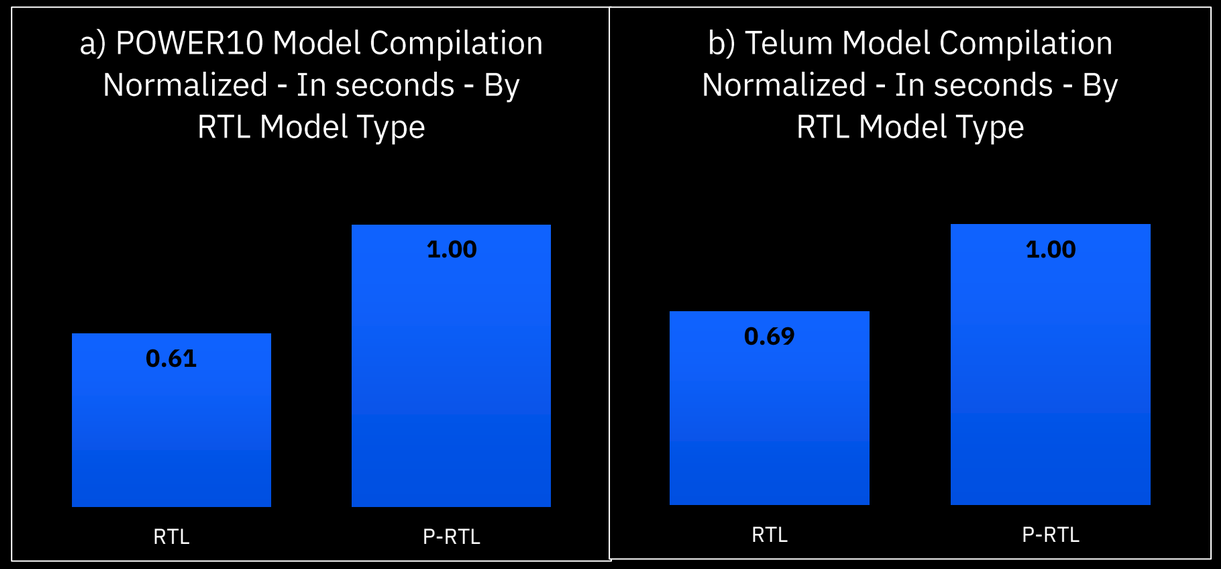


Figure 3 | a) POWER10 b) Telum - *Model Compilation Time*

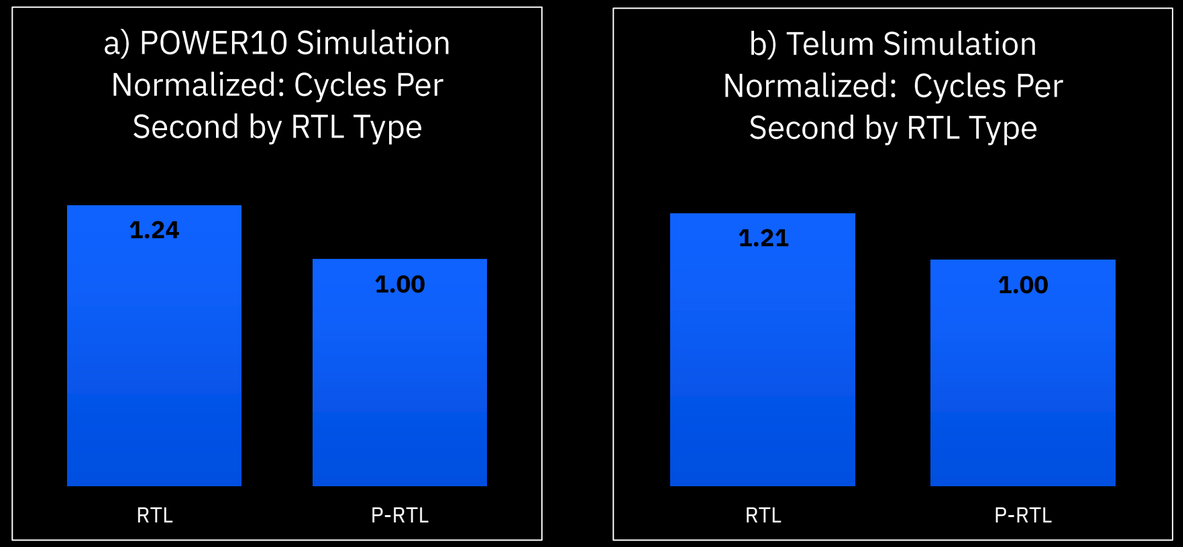
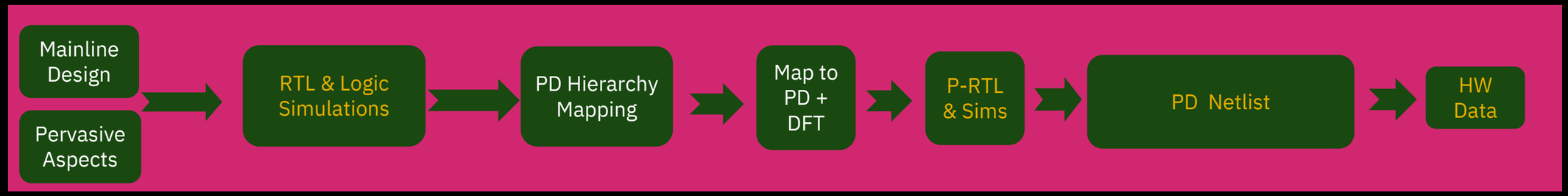


Figure 4 | a) POWER10 b) Telum - *Cycles Per Second*

AOD | Key Differentiators

(1) RTL & Simulation Debug	(2) RTL & Post–Aspect–Inserted RTL (P-RTL) Simulation and Emulation Debug	(3) Logic & Physical Design (PD) Debug	(4) Post-Silicon Hardware Logic Debug



- ✓ RTL level AOD flow is notably different from existing methods for netlist level aspect insertion (which causes notable overheads for verification and debug)
- ✓ An automated logical-to-physical hierarchy mapping enables physical designers to leverage this without dependence on logic engineers
- ✓ Aspect insertion is done in RTL, debugging easier than netlist flows (where synthesis name mangles the RTL)

Cycle Simulation v/s Traditional Techniques

Event Simulation

- Delay and timing information included
- Can handle both synchronous and asynchronous circuits
- Changes at time t triggers an event at t + Δt
- Speed of simulation depends on model activity
- Driven by UVM/System Verilog test benches
- Large designs this can be slow
- VCS, Xcelium, Model-Sim

Cycle Simulation

- Delay and timing information ***not*** included
- Can handle only synchronous circuits
- Assumes switching is synchronous and evaluates all gates each cycle
- Speed of simulation doesn't depend on model activity, so this scales better for larger designs
- Driven by faster C++ testbenches
- Large designs this can be faster
- MESA (IBM)

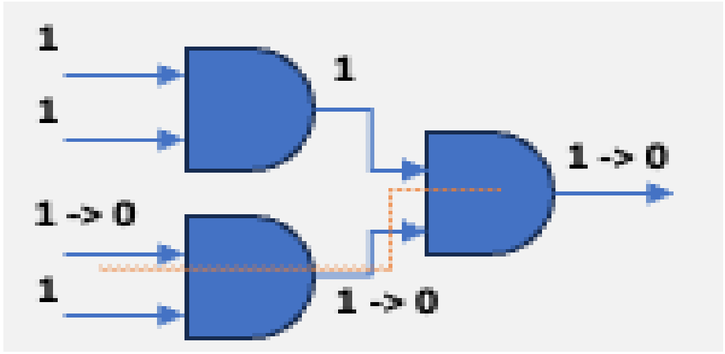


Figure 6 | Event Simulation – Changes triggers events down the line

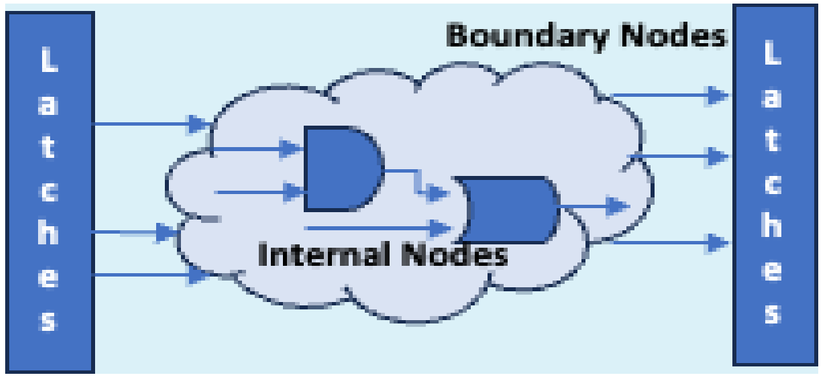


Figure 7 | Cycle Simulation - Boundary evaluated every cycle

XTB | Cross Test-Bench | Distributed Co-Simulation Platform

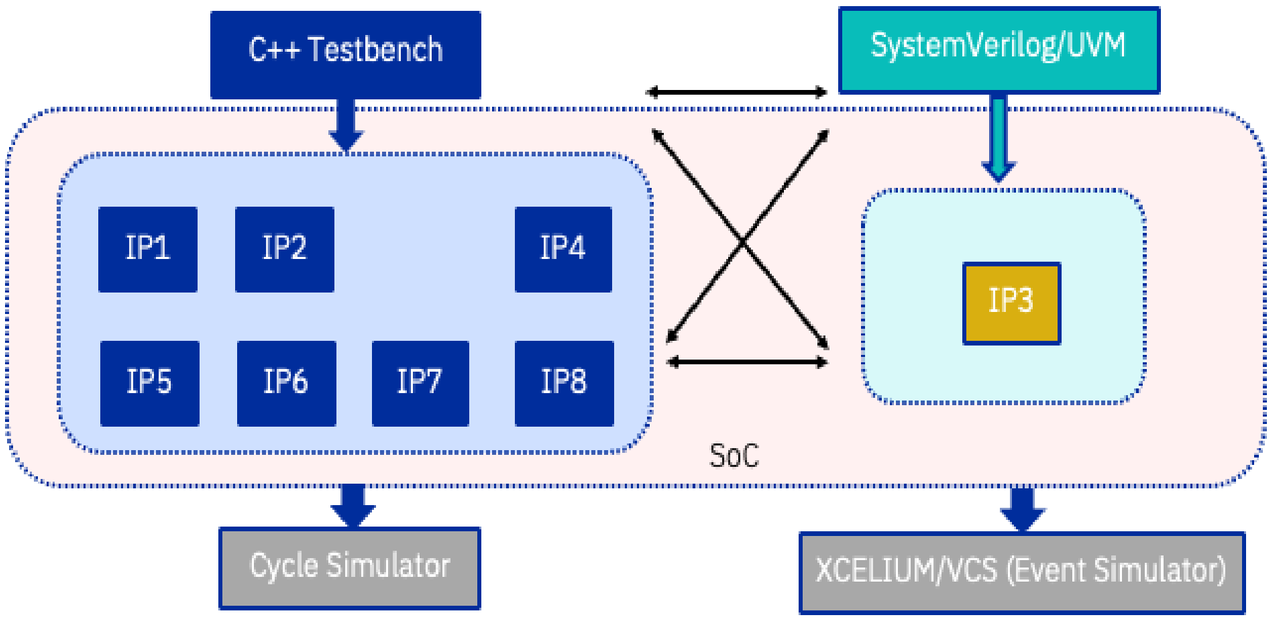


Figure 8 | XTB allows IPs to run in Individual Simulators while allowing to Exchange Values

DAC 2024 [Link]

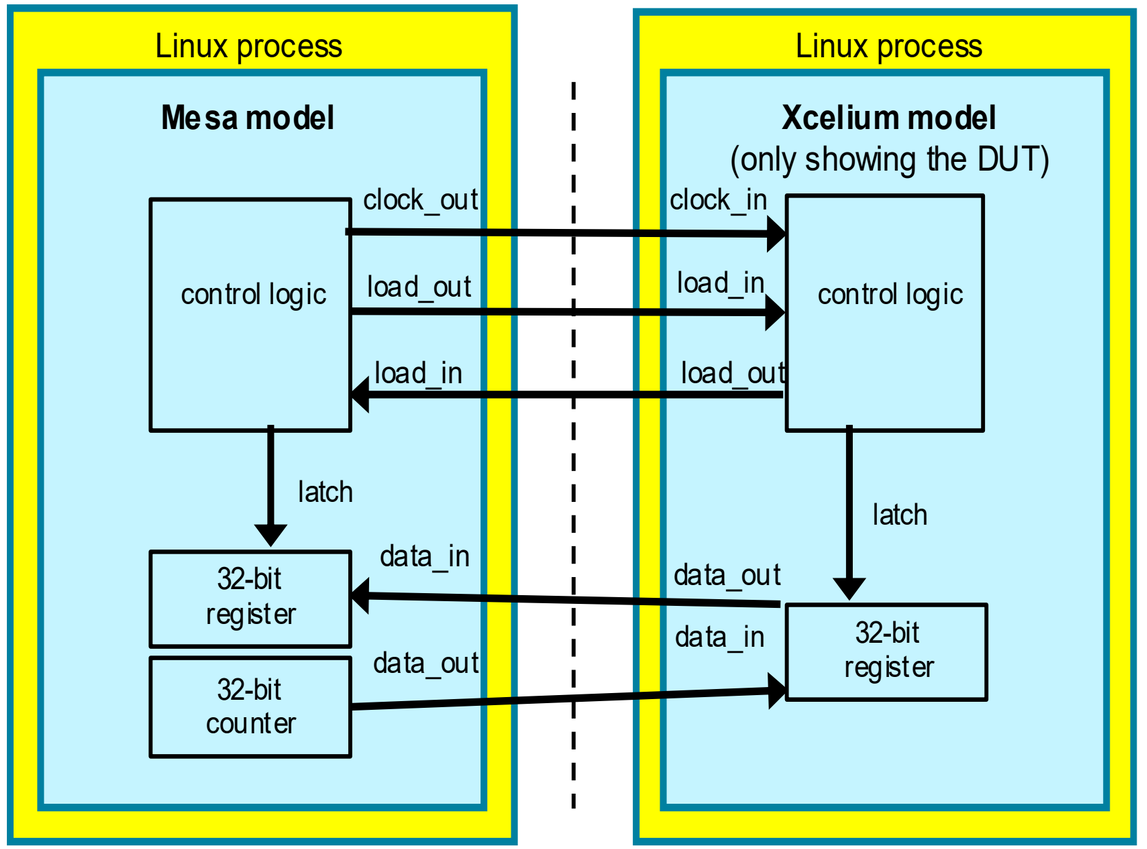
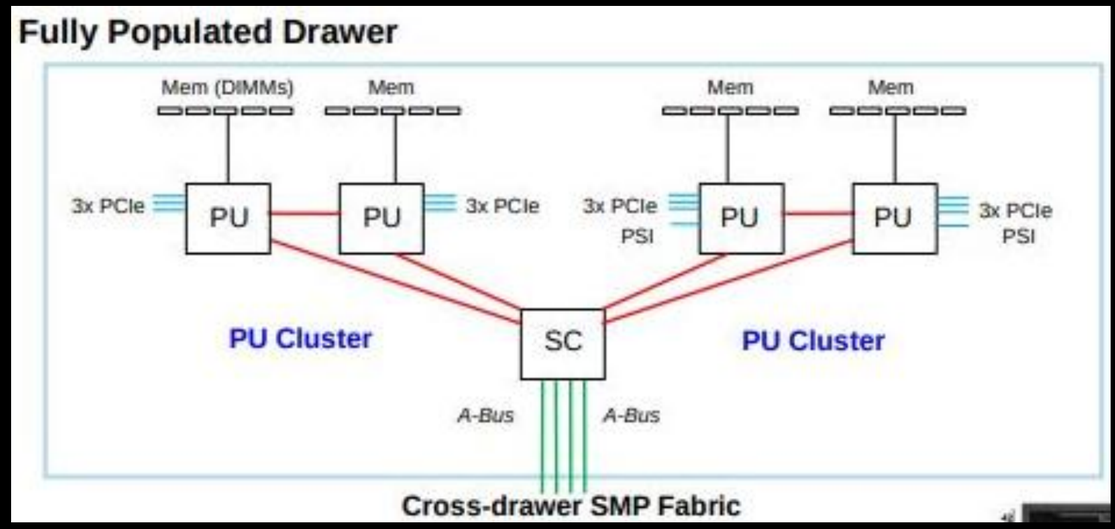


Figure 9 | Example of MESA & Xcelium Co-simulation

- Supports DUT-2-DUT, TB-2-DUT, and TB-2-TB communication channels
- Also target a hierarchical signal/wire deep within the design

XTB | Hybrid Solution

- Leverages advantages of both methodologies
- Event simulation at unit level to get timing accurate verification
- Cycle simulation at chip level for high speed and coverage
- XTB sim gets both by allowing timing accurate unit models to simulate with chip cycle models
- Use-case was IBM's POWER9 interfacing with NVIDIA's GPU



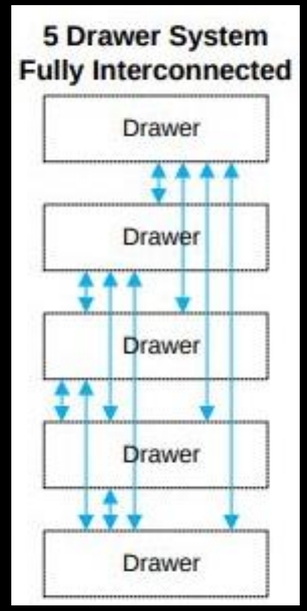


Figure 10 | IBM leverages cycle simulation to simulate server processors and achieve ~100% coverage

Outline

- High performance design context
- IBM methodology for verification
- AI/ML techniques
- IBM debug platform

Accelerating Coverage Driven Generation Verification

- ✓ Replace DUV simulation with ML model | Capture relation between test-templates and coverage
- ✓ Instead of using ML model as main ingredient of CDG | Use to accelerate DFO search algorithm

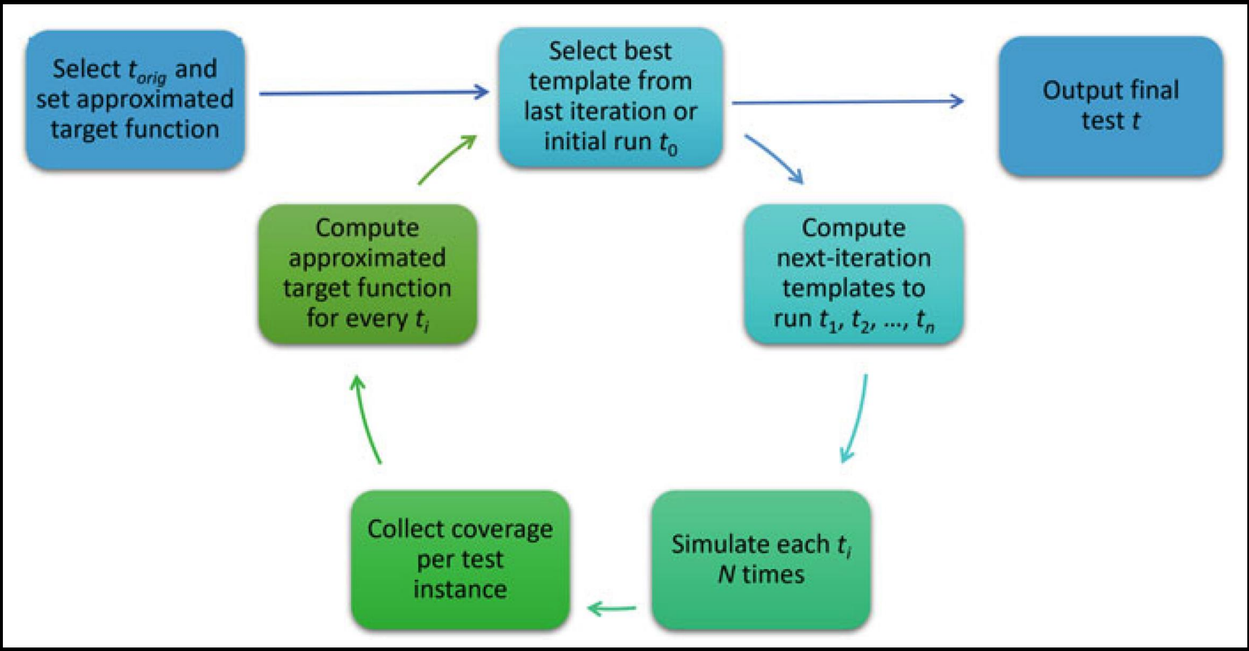


Figure 11 | Automatic feedback loop from coverage data to stimuli generation

Event name	Before CDG (1,000,000 sims)		Sampling phase (210 tests x 100 sims each)		Optimization phase (25 iteration x 12 tests x 100 sims)		Running best test (15000 sims)	
	# hits	hit rate	# hits	hit rate	# hits	hit rate	# hits	hit rate
byp_reqs01	865,000	86.500%	18,109	86.233%	24,957	83.190%	14,478	96.520%
byp_reqs02	284,800	28.480%	10,015	47.690%	19,569	65.230%	13,038	86.920%
byp_reqs03	25,600	2.560%	4,425	21.071%	13,251	44.170%	9,608	64.053%
byp_reqs04	2,000	0.200%	3,109	14.805%	11,232	37.440%	8,409	56.060%
byp_reqs05	300	0.030%	2,558	12.181%	10,339	34.463%	7,989	53.260%
byp_reqs06	0	0.000%	2,052	9.771%	9,492	31.640%	7,536	50.240%
byp_reqs07	0	0.000%	1,598	7.610%	8,485	28.283%	7,026	46.840%
byp_reqs08	0	0.000%	1,242	5.914%	7,424	24.747%	6,387	42.580%
byp_reqs09	0	0.000%	903	4.300%	6,268	20.893%	5,559	37.060%
byp_reqs10	0	0.000%	640	3.048%	4,957	16.523%	4,537	30.247%
byp_reqs11	0	0.000%	389	1.852%	3,494	11.647%	3,401	22.673%
byp_reqs12	0	0.000%	193	0.919%	2,148	7.160%	2,245	14.967%
byp_reqs13	0	0.000%	90	0.429%	1,096	3.653%	1,180	7.867%
byp_reqs14	0	0.000%	33	0.157%	450	1.500%	462	3.080%
byp_reqs15	0	0.000%	6	0.029%	125	0.417%	127	0.847%
byp_reqs16	0	0.000%	0	0.000%	24	0.080%	15	0.100%

Figure 12 | Optimization techniques to achieve Coverage Driven Generation (CDG)

AI/ML Assisted Verification Closure | Results

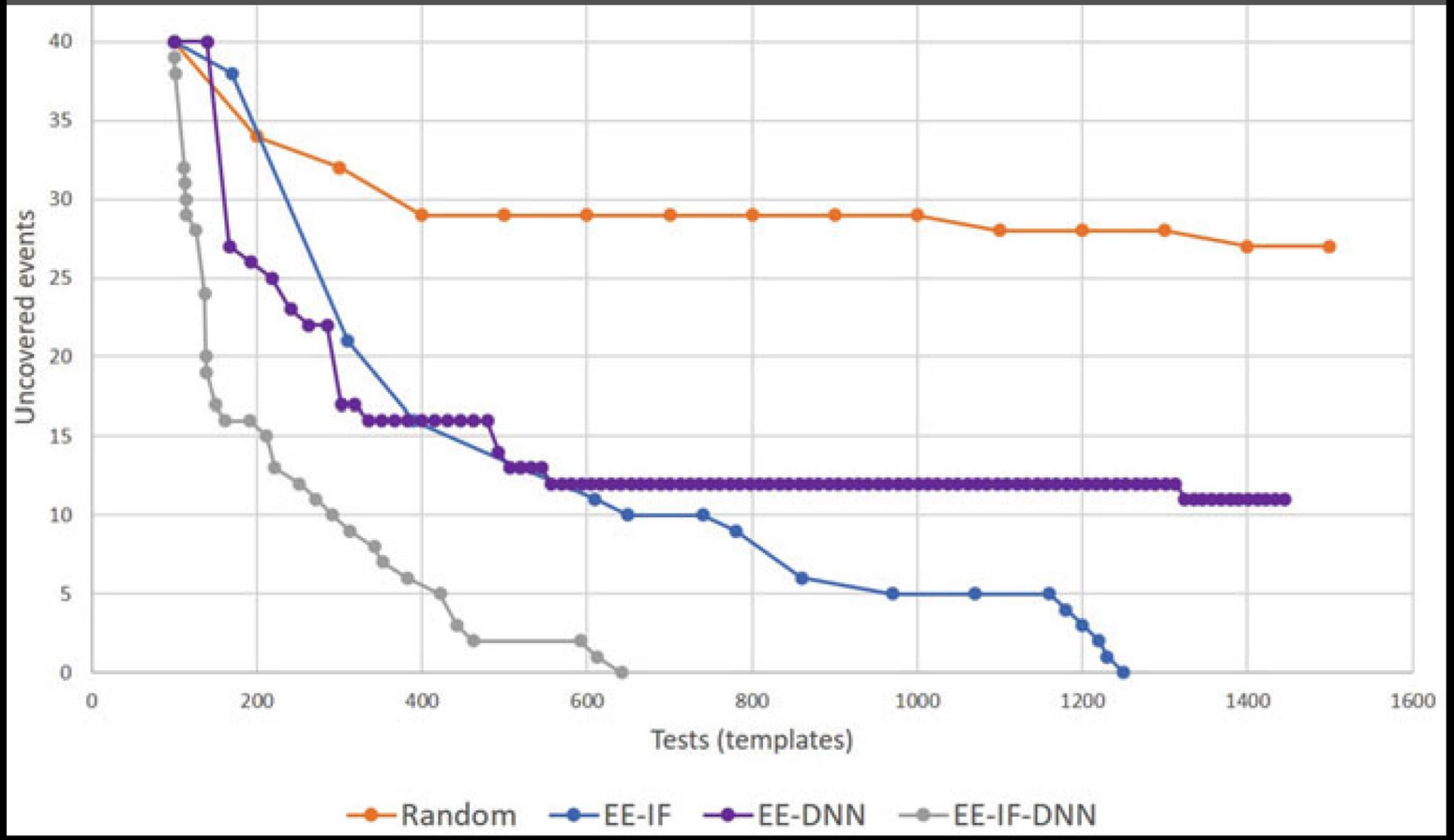
1. Random—Generate random tests to cover events

2. EE-IF—Use the event after events implicit filtering

3. EE-DNN—DNN to eliminate events without implicit filtering

4. EE-IF-DNN—Use the DNN accelerated search algorithm

Figure 13 | Comparing the different algorithms, starting from the same initial 100 random test templates



ISQED'23 [Link]

Outline

- High performance design context
- IBM methodology for verification
- AI/ML techniques
- IBM debug platform

NEXA | Collaborative Hardware Logic Debug Platform

Figure 14 | Single Consistent Platform

RTL Debug | Logic Simulation Debug | Logic-Physical Debug | Post-Silicon Debug

Stage 1: RTL and Simulation Debug

Stage 2: RTL and Post-Aspect-Inserted RTL (P-RTL) Simulation and Emulation Debug

Stage 3: Logic and Physical Design (PD) Debug

Stage 4: Post-Silicon Hardware Debug

Mainline Design

Non-Mainline Aspects

RTL & Logic Simulations

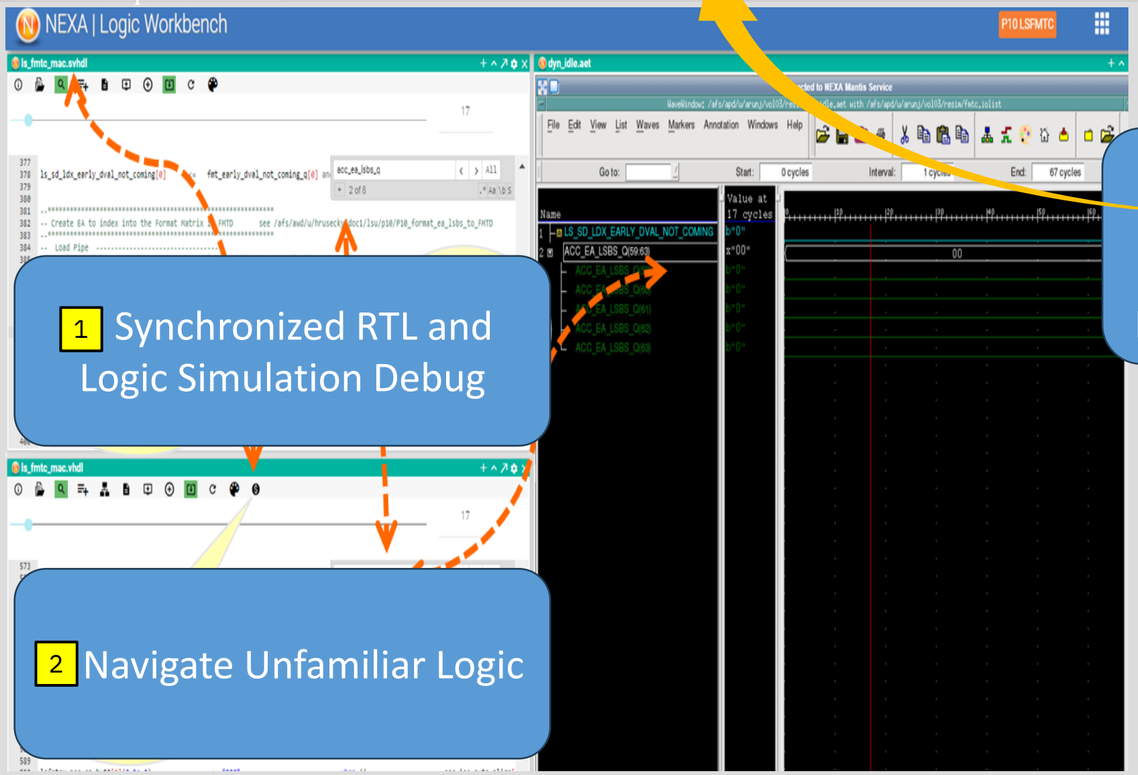
PD Hierarchy Mapping

Map to PD + DFT

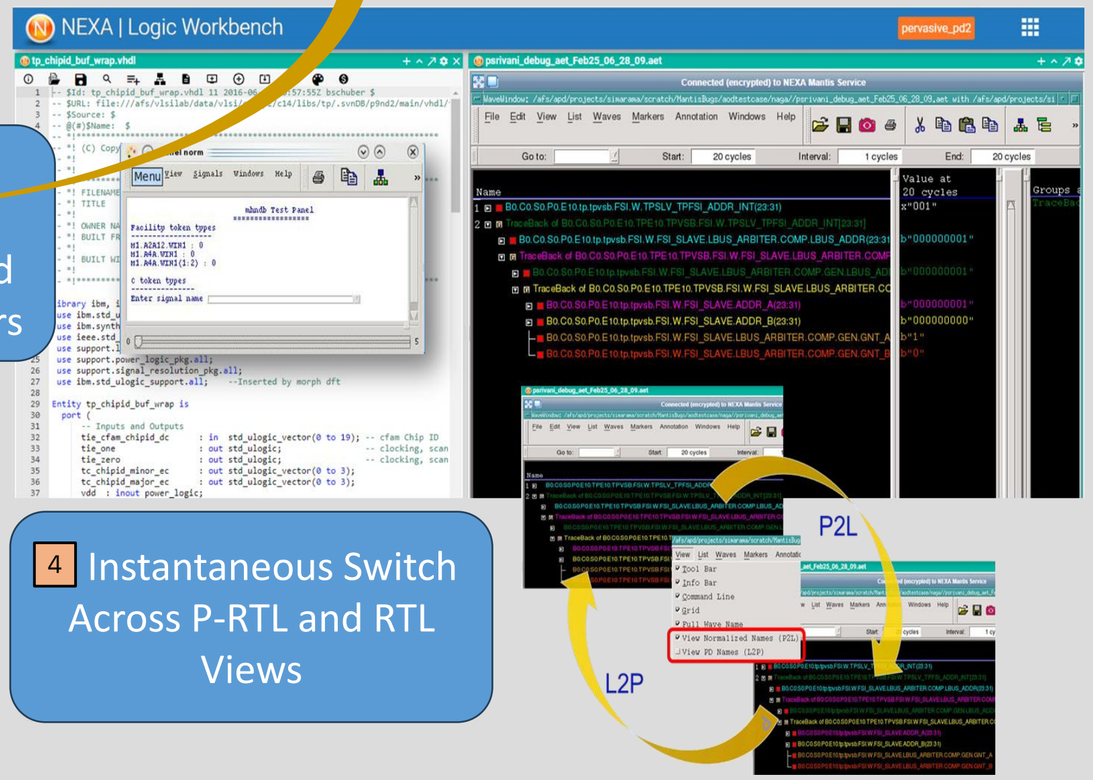
P-RTL & Simulations

PD Netlist

HW Data



3 Collaborative Across Non-Mainline and Mainline Engineers



4 Instantaneous Switch Across P-RTL and RTL Views

NEXA | Use-cases

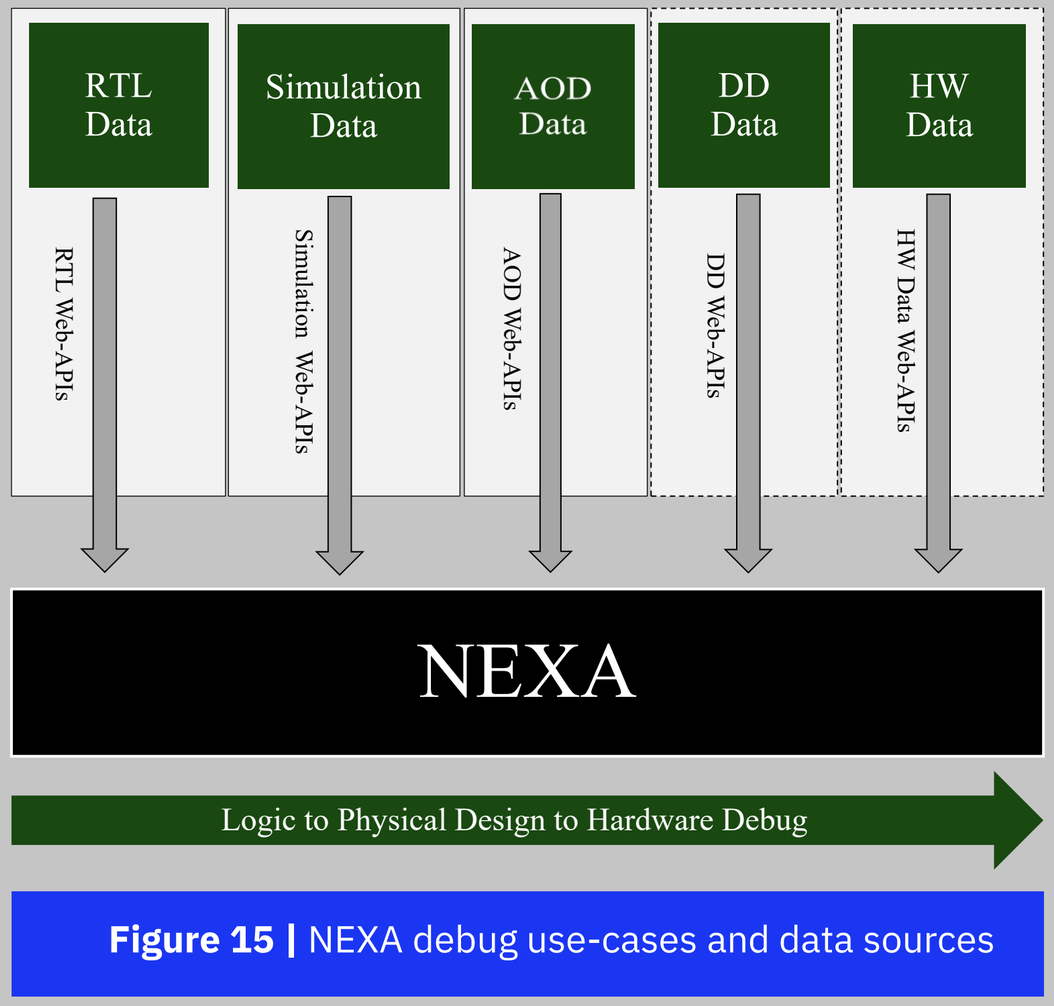


Figure 15 | NEXA debug use-cases and data sources

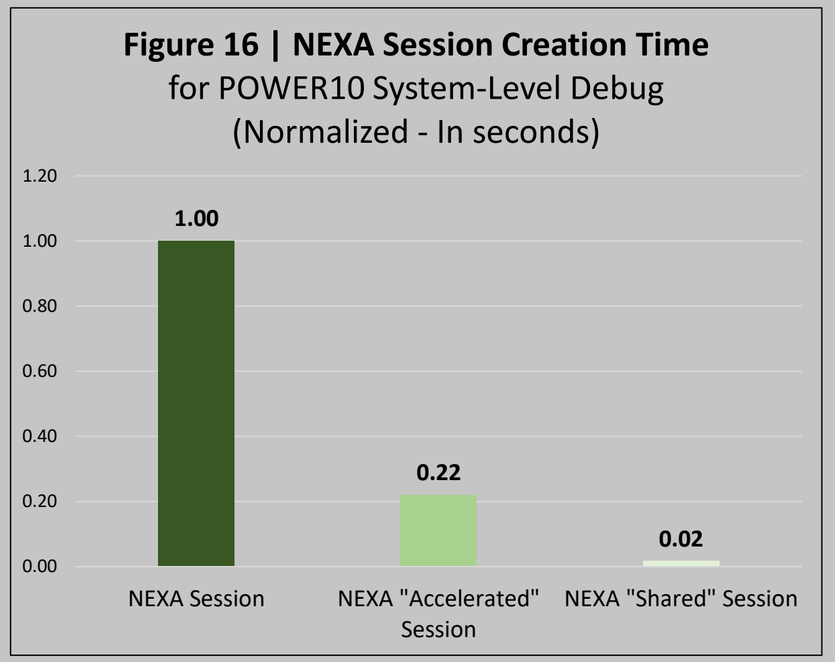


Figure 16 | NEXA Session Creation Time for POWER10 System-Level Debug (Normalized - In seconds)

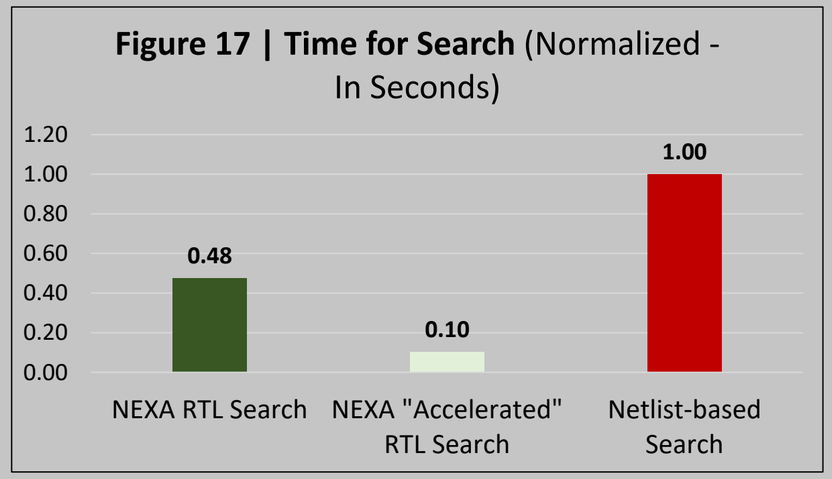


Figure 17 | Time for Search (Normalized - In Seconds)

Hardware | NEXA LabReplay

- Create “much expanded” data in the lab for high-performance hardware debug

- Shift right principle of Designer Level Verification
- Perform “replay” on simpler model of hardware
- New application of X-state simulation

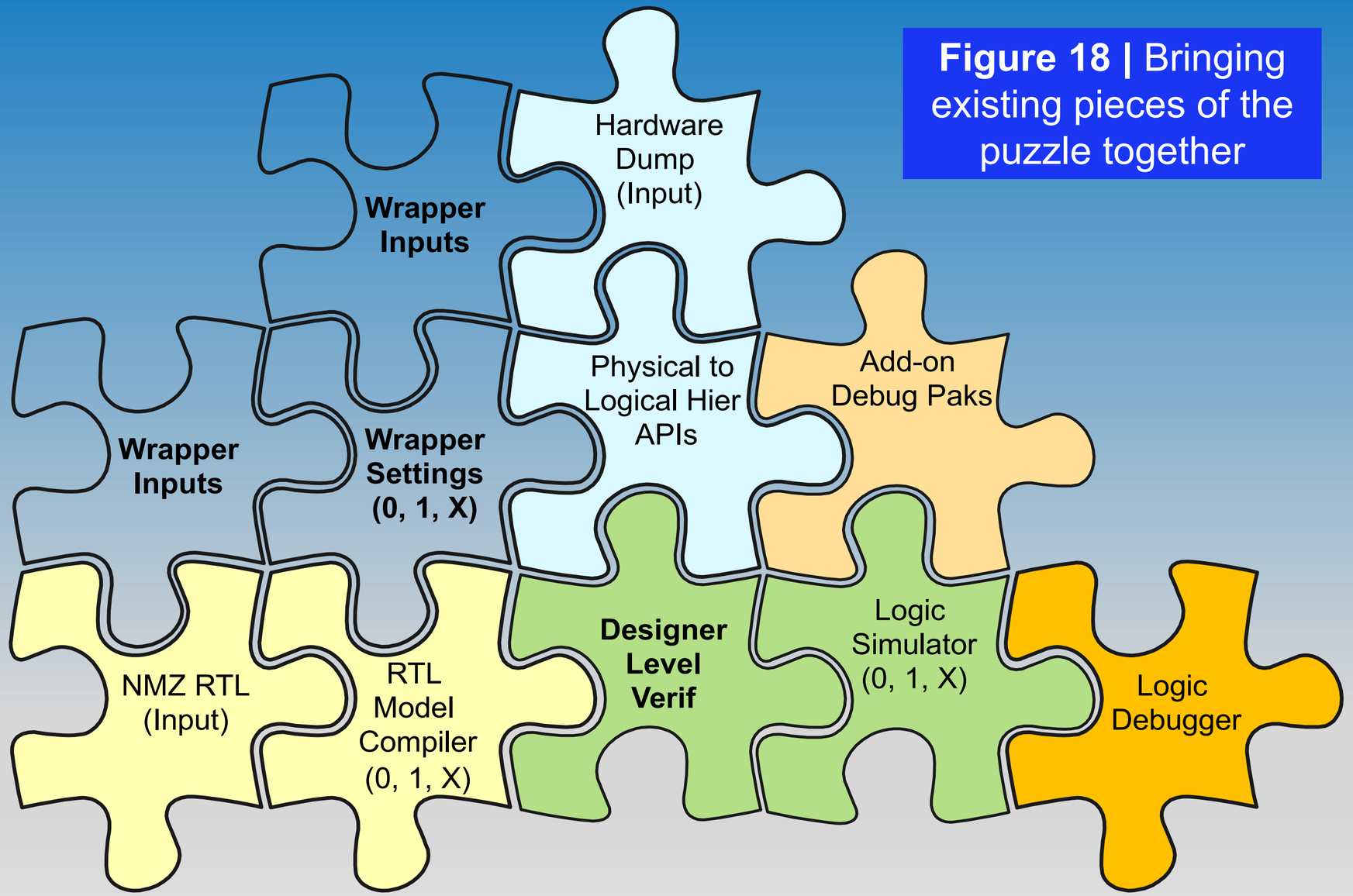


Figure 18 | Bringing existing pieces of the puzzle together

Summary

- New functions across EDA helps faster design, verification and debug closure of IBM hardware
- AI/ML techniques augments traditional design closure, and delivers new experiences
- Single-team mindset key for sustainable innovation in next generation of high-performance hardware

References

- [1] A. Joseph, P. M. Jacob, M. Klein and W. Roesner, "Efficient Aspect Verification and Debugging of High-Performance Microprocessor Designs," in IEEE Design & Test, vol. 41, no. 3, pp. 36-46, June 2024, doi: 10.1109/MDAT.2024.3363085. [Link]
- [2] Rao et al., "POWER10™: A 16-Core SMT8 Server Processor With 2TB/s Off-Chip Bandwidth in 7nm Technology," 2022 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2022, pp. 48-50. [Link]
- [3] Jacobi, "Real-time AI for Enterprise Workloads: the IBM Telum Processor," 2021 IEEE Hot Chips 33 Symposium (HCS), Palo Alto, CA, USA, 2021, pp. 1-22.
- [4] IBM Telum [Link]
- [5] Pundir et al., “A Distributed Co-Simulation Environment and its Application in HW-FW Verification”, DAC 2024 [Link]
- [6] Joseph et. a.l. “NEXA | Cloud Native Platform for Collaborative Hardware Logic Design in Step-wise Refinement Implementation Flows”, DAC 2021 [Link]
- [7] Bavikadi et al., "A Survey on Machine Learning Accelerators and Evolutionary Hardware Platforms," in IEEE Design & Test, vol. 39, no. 3, pp. 91-116, June 2022 [Link]
- [8] Avi et al. "Lightning Talk: Functional Verification–Past, Present, and Future Challenges and Opportunities." DAC 2023 & ISQED’23 [Link]