2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SAN JOSE, CA, USA
FEBRUARY 24-27, 2025

Accelerating Pre-Silicon Verification Coverage with Transaction Sequence Modeling

Jayanth Raman, Jackson Wydra, Ximin Shan, Rahul Krishnamurthy, Michael Yan, Phyllis Hsia, Vikram Narayan, Samir Mittal

micron™

accellera SYSTEMS INITIATIVE

Background and Methodology

Background

- Design Verification (DV) is a crucial phase in the lifecycle of integrated circuit product development.
- DV primarily employs simulation-based methodologies
  - Popular. But time and resource intensive.
- Coverage is a standard DV metric. Quantifies the extent of testing.
- Universal Verification Methodology (UVM) is a standardized framework used in DV.
- Transaction sequences in UVM generate and manage the flow of data items, or transactions. It defines the simulation in a test case.

Transaction Sequence Modeling – Main Idea

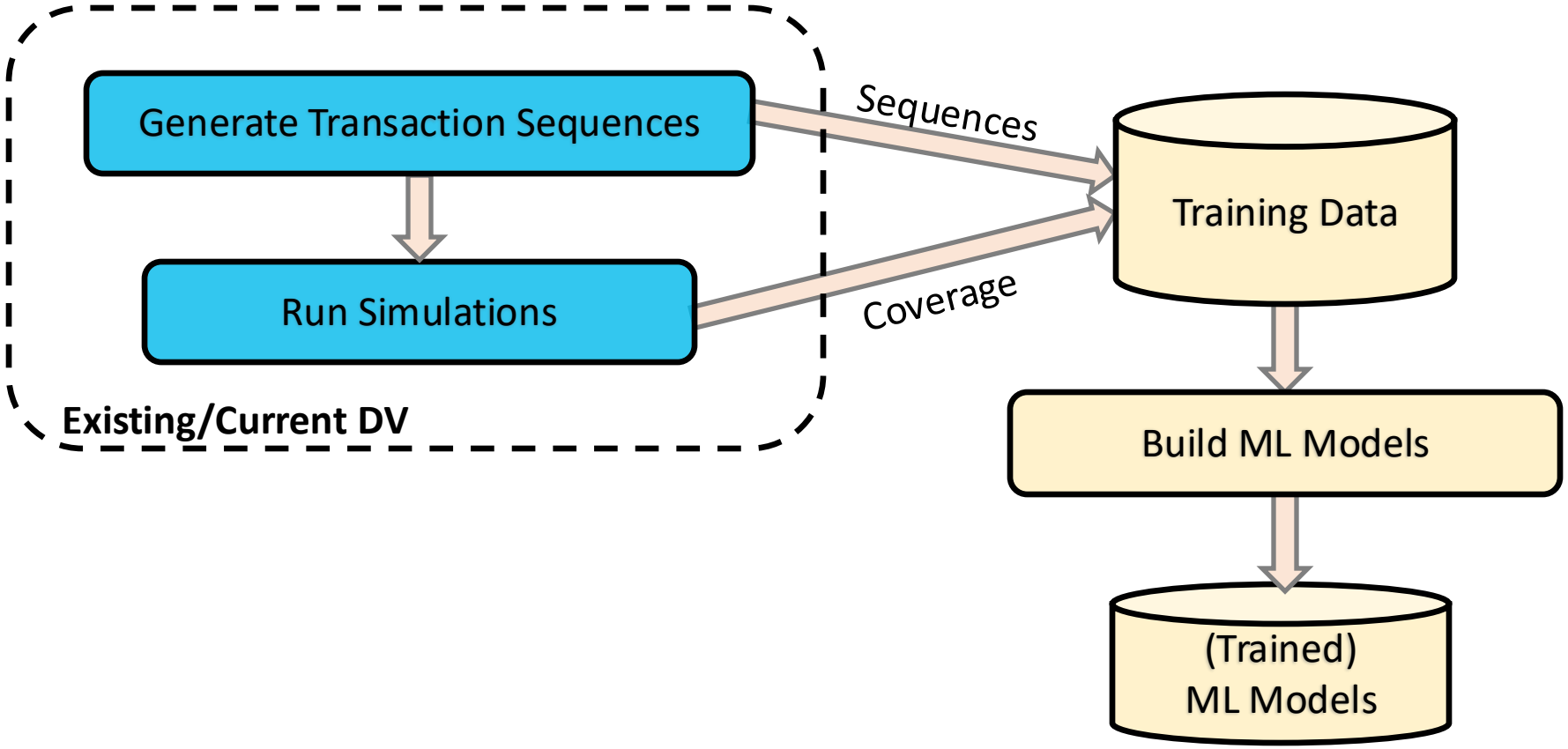
- As part of running a test case, transaction sequences are generated.
- Determine if a set of transactions for a test case simulation is likely to increase coverage or not.
- If the set of transactions for a test case are not likely to increase coverage, skip the simulation to shorten verification time.

Methodology

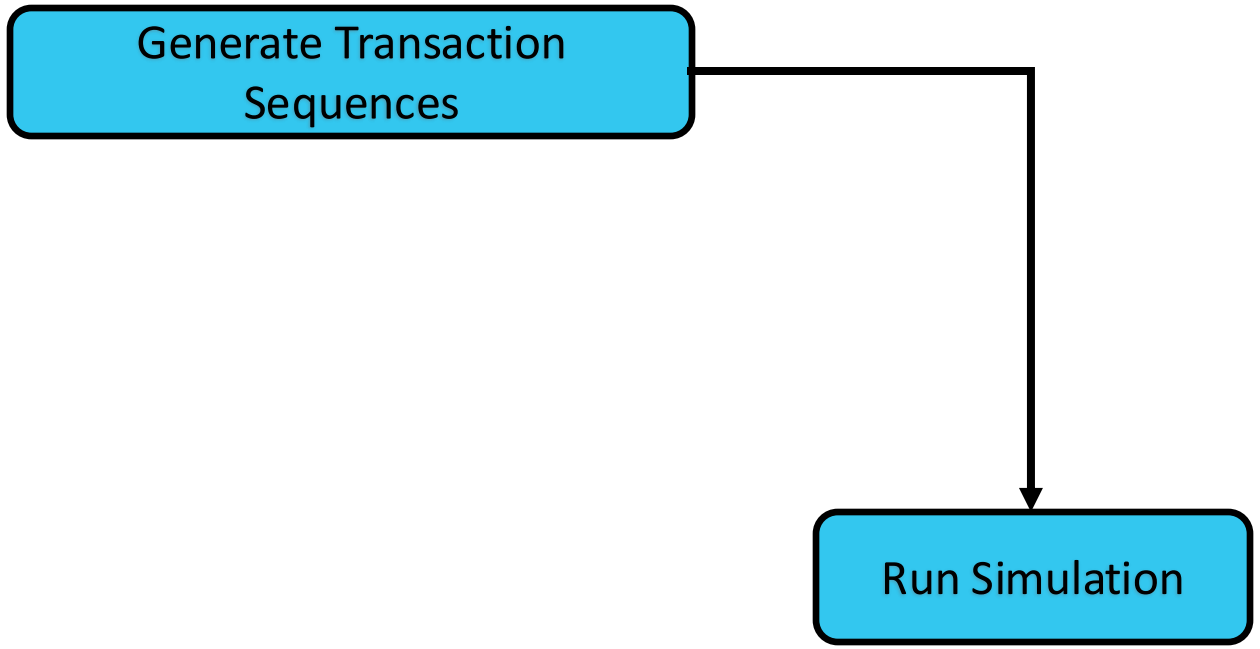
Two phases:

1. Train: Run simulations to collect data, then use it to train and store ML models.
2. Infer: (1) Generate transaction sequences as part of running a test case. (2) Skip the simulation if ML models predict the sequences are unlikely to increase coverage.

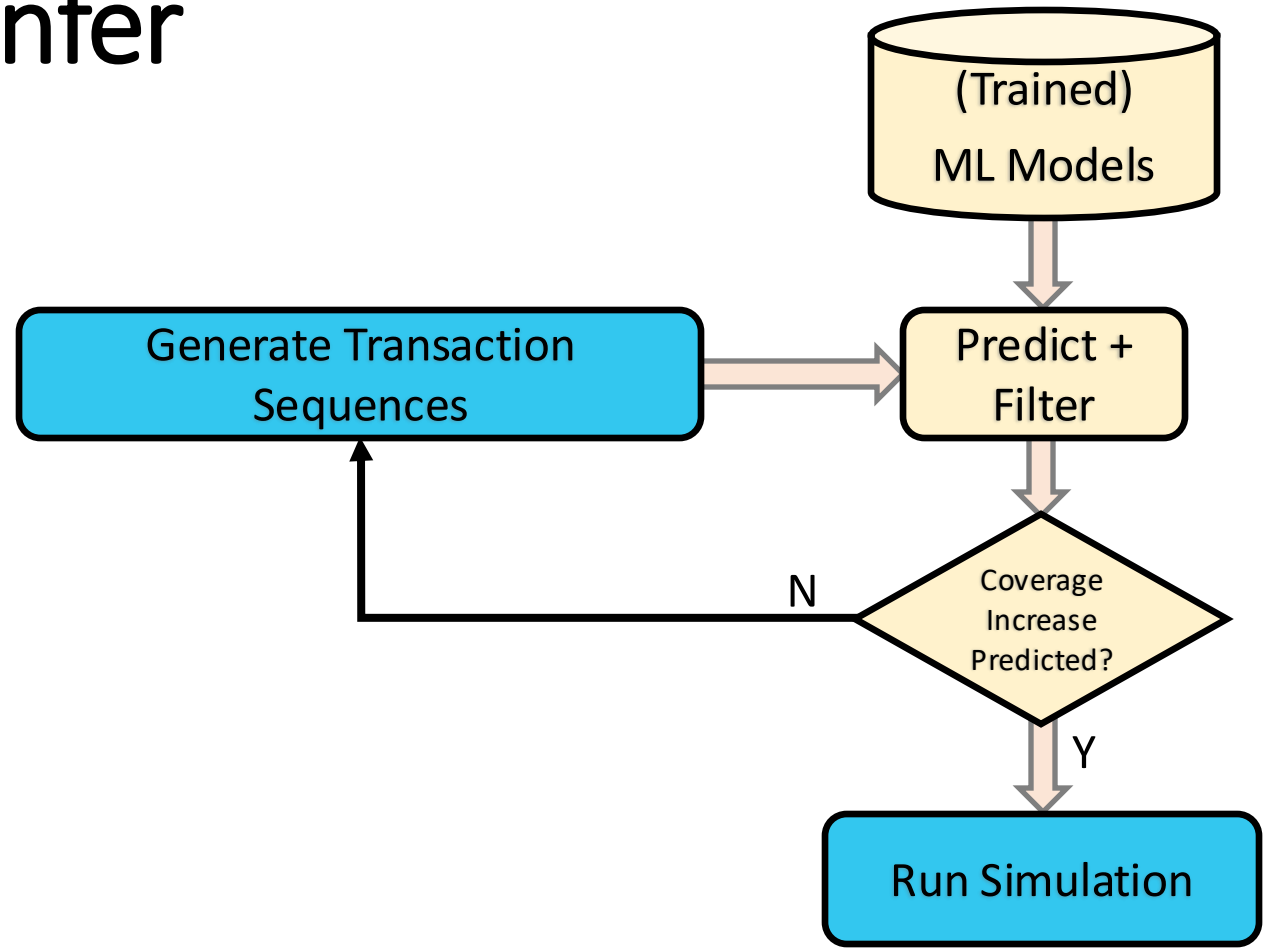
Phase I – Train



Normal DV Flow



Phase II – Infer

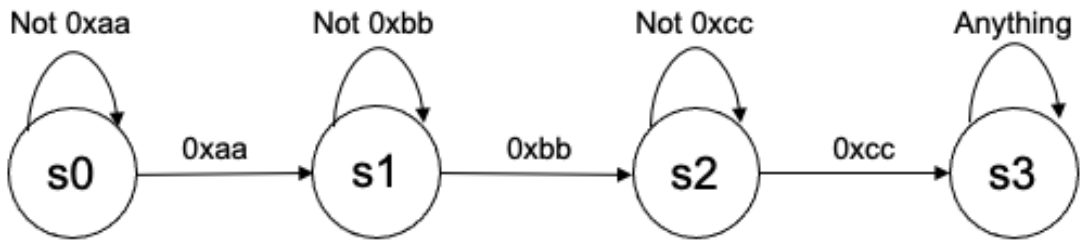


Results

accellera
SYSTEMS INITIATIVE

Lock DUT

- **Lock** Design-Under-Test (DUT) is an RTL design that accepts a sequence of code and only unlocks if the sequence matches a predefined pattern.
- It's a significant coverage challenge for larger bit-widths of the code.
  - E.g., for a sequence length of 3 and 8 bits in the code, the probability of unlocking is 1 / $(256)^3$ (or one in about 16 million).



Lock DUT: Coverage Acceleration Procedure

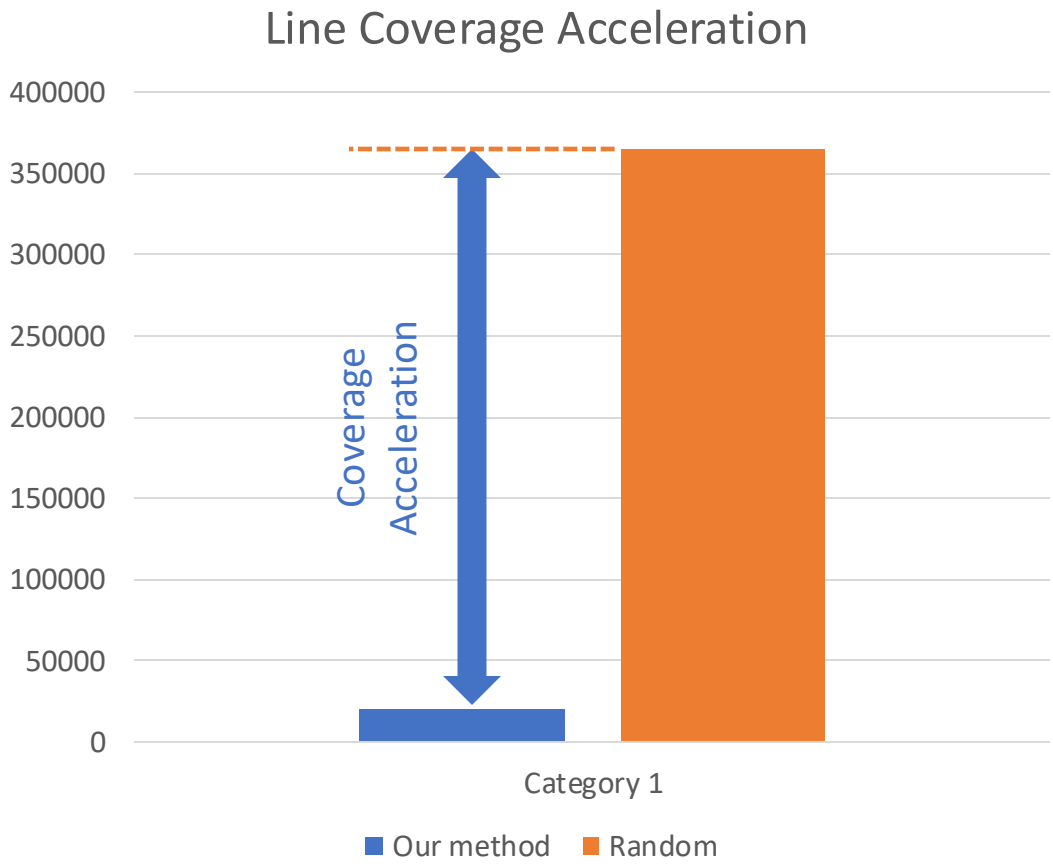
- Phase I: Model was trained to predict coverage of the line hitting the first code.
- Phase II: Trained model used to only keep sequences that were predicted to hit that line.
- Simulations were run on only these filtered sequences.

Lock DUT: Model Training Setup

- Code length = 8 bits (e.g. 0x42)
- Sequence length = 8 (e.g. [0x12, 0xaa, 0x43, 0xbb, …, 0xcc, …])
- ML Model Input = 8 x 1 vector of integers (range: 0 to 255)
- ML Task = Classification = 1 if state s1 is reached, 0 otherwise
- ML Model = standard XGBoost, LightGBM
- Number of samples = 10K simulations.
- Training/validation split – 70-30.

Lock DUT: Coverage Acceleration Result

- **Random simulations:** 366K simulations on average.
- **Our method**: about 20K simulations on average.
- **Coverage acceleration** = 18x



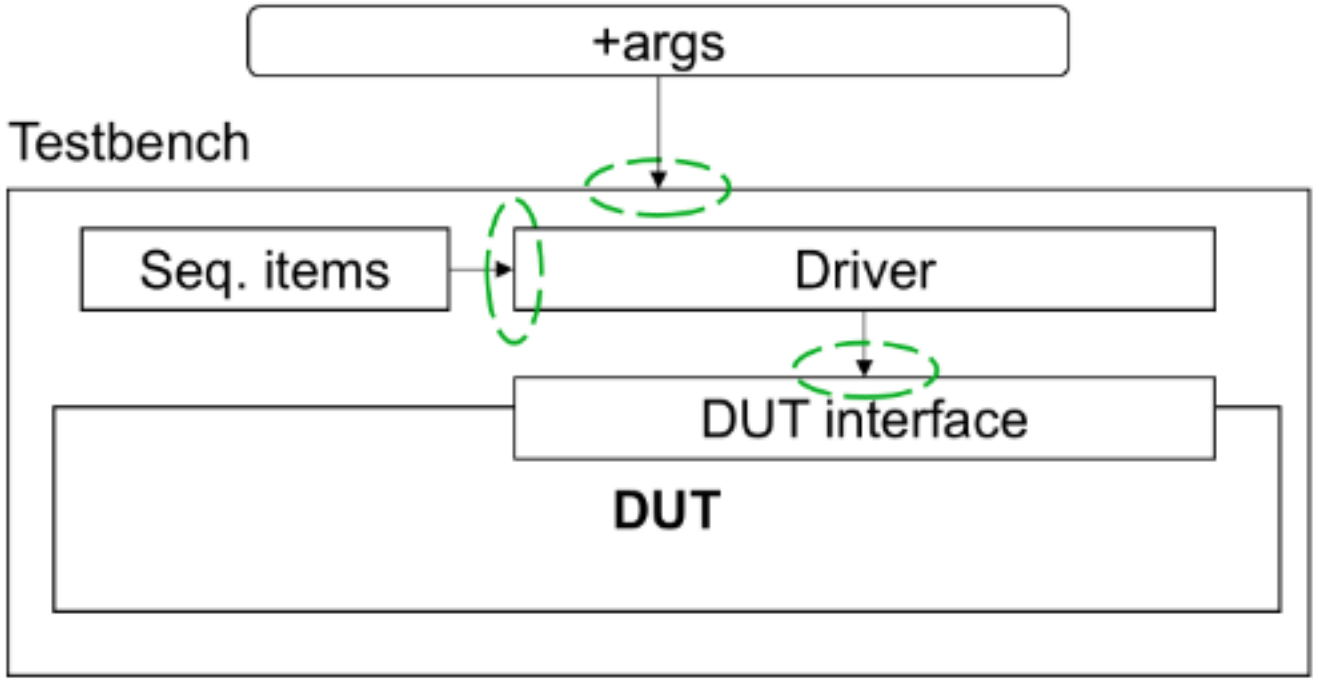
Cache DUT

- Micron internal RTL design.
- Cache Subsystem processes requests received from a Management Subsystem.
- It supports encryption and decryption.

Terminology and Notes

- Design code is in Verilog.
- Design code is composed of modules.
- Instance = instantiation(s) of a module.
- An (instance, line-number) is a unique coverable line.
- Line coverage: a line is covered if it was executed by a test simulation.
- Covergroup is a construct for functional coverage. A covergroup is composed of one or more cover points which in turn are composed of one or more cover bins.

Testbench Instrumentation



Transaction Sequences

Write Transaction Sequence

Timestamp	BufferID	Key	X	Y
—	—	—	—	—
—	—	—	—	—
—	—	—	—	—

Read Transaction Sequence

Timestamp	BufferID	Key	X	Z
—	—	—	—	—
—	—	—	—	—

Merged Transaction Sequence

Cmd	Timestamp	BufferID	Key	X
0	—	—	—	—
1	—	—	—	—
0	—	—	—	—
0	—	—	—	—
1	—	—	—	—

Flattened binary vector.

Cache DUT: Model Training Setup

- One model per line or cover bin.
- Excludes easy to hit lines (>90% of tests hit these lines) and very hard to hit (<10%).
- Model Input: Binary vector of length ~22k.
- Model Task: Binary Classification: 1 if line/bin is hit, 0 otherwise.
- ML Model: XGBoost/LightGBM/DummyClassifier/…

Cache DUT Results

Line Coverage

- Models AUC of lines shown grouped by module.
- Many lines are modeled well (high AUC). Others are challenging.

	Module	Mean Best AUC	AUC Range	Unique LineN	Unique Inst.	Uniq (Inst., LineN)
Poor	Module 1	0.50	0.50—0.51	1	3	3
	Module 2	0.51	0.51—0.51	1	1	1
	Module 3	0.51	0.50—0.52	25	1	25
	Module 4	0.51	0.50—0.52	2	2	3
	Module 5	0.51	0.51—0.51	3	1	3
	Module 6	0.56	0.50—0.81	1	48	48
	Module 7	0.64	0.50—0.99	4	5	7
	Module 8	0.70	0.50—0.99	3	5	15
	Module 9	0.75	0.50—1.00	7	20	50
Good	Module 10	1.00	1.00—1.00	1	12	12
	Module 11	1.00	1.00—1.00	1	231	231
	Module 12	1.00	1.00—1.00	7	1	7
	Module 13	1.00	1.00—1.00	7	32	224
	Module 14	1.00	1.00—1.00	1	3	3
	Total			**64**	**365**	**632**

* LineN = Line-Number.

Area Under the Curve (AUC) is a standard ML metric. An AUC of 1 indicates perfect classification, while an AUC of 0.5 suggests random guessing.

Cache DUT Results

Group Coverage

- Models AUC of cover bins shown grouped by Covergroup.
- A Covergroup contains multiple coverpoints, contains multiple bins.
- About 28% of the bins are have high prediction performance.

Covergroup	Num Coverpoints	Num Bins	Mean AUC	Min AUC	Max AUC
Covergroup 1	1	5	0.80	0.50	1.00
Covergroup 2	8	8	0.50	0.48	0.52
Covergroup 3	3	11	0.78	0.50	1.00
Covergroup 4	9	10	0.51	0.49	0.53
Covergroup 5	1	5	0.80	0.50	1.00
Covergroup 6	2	2	0.51	0.51	0.52
Covergroup 7	1	2	0.50	0.49	0.51
Total	25	43			

Summary

Summary

- Transaction sequences can be used as input to train machine learning models to predict code and functional coverage.
  - Lock DUT.
  - Cache DUT.
- Coverage acceleration was demonstrated by using the transaction sequence ML models.
- More work needed
  - Improve methodology on the Cache DUT.
  - Apply this methodology on other DUTs.

Questions?