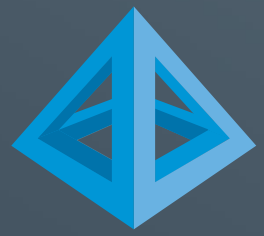


BREKER™ The Leader in Portable Stimulus

AI-Based Test Content Synthesis

An Example of AI Complexity Management

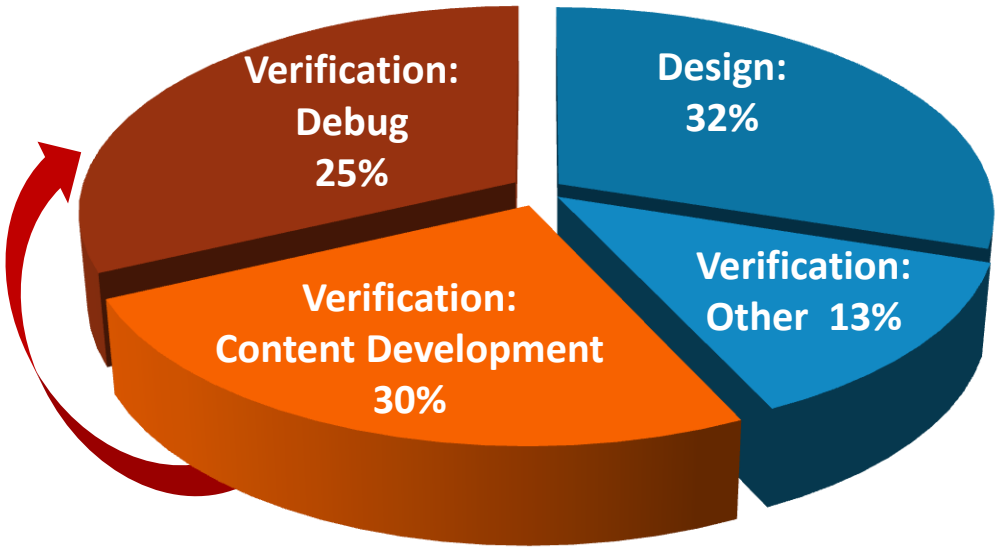
Dave Kelf, CEO, Breker Verification systems

DVClub Europe, November 2021

Test Content Composition Efficiency is Critical

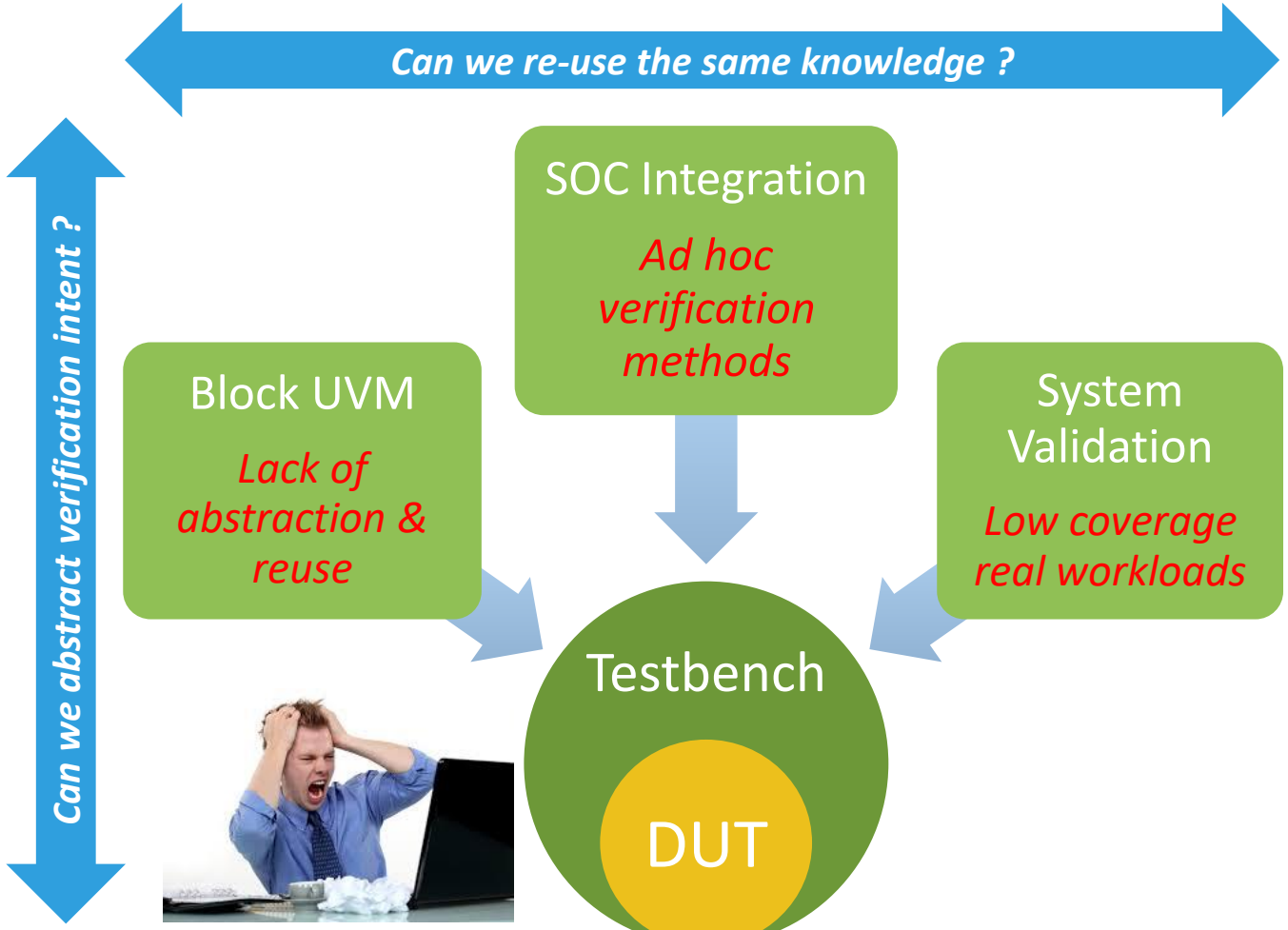
Project Resource Deployment

Source: Wilson Research 2020



Test development drives debug

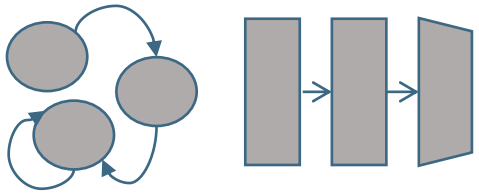
Complex tests hard to get right



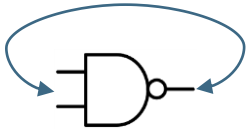
Why? Resource Intensive Test Content

Transformational Approach: Test Content Synthesis

Design Synthesis

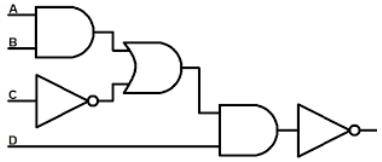


Constrain

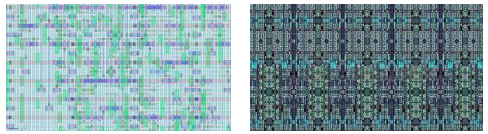


Timing/Area Constraints

Synthesize



Optimize



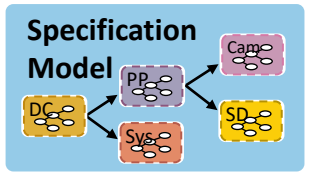
Describe intent

Specify goals

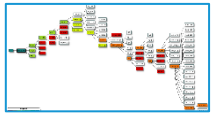
Generate implementation

Map to platform

Test Suite Synthesis

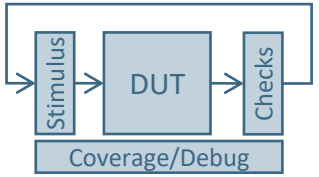


Constrain



Coverage Constraints

Synthesize



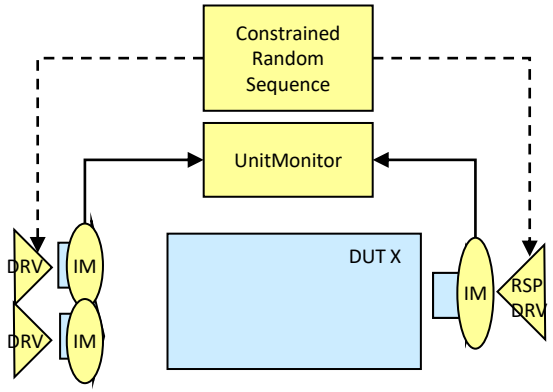
Optimize

Breker Core Technology

- Executable Specification
- 3D Coverage Closure
- Synthesizable VerificationOS

AI Planning Algorithms

IBM View: Limits of Constrained Random Testing

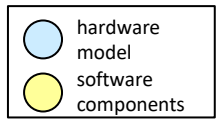


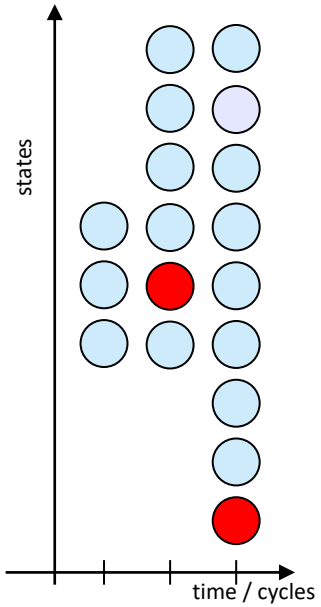
constrained random transactions are well-suited for logic with broad state space with low sequential depth
"shotgun principle"

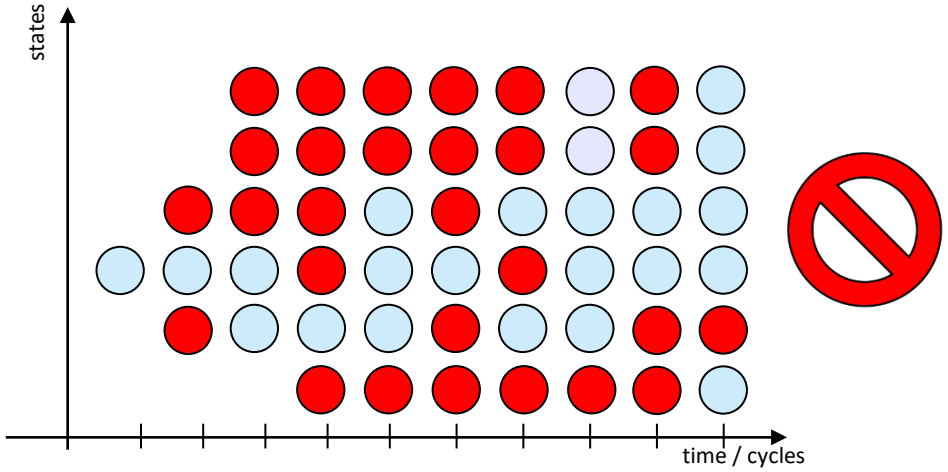
… but they are weak in describing sequences (with random decisions) for logic with sparse state space and high sequential depth

Courtesy

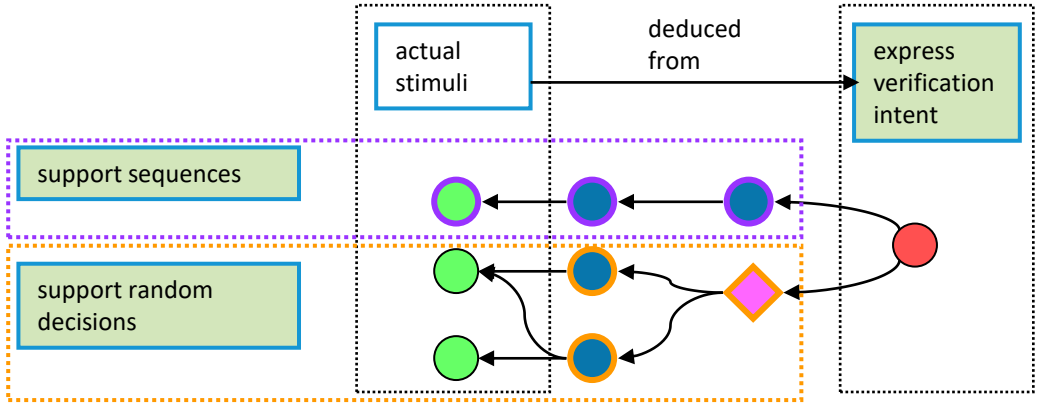
IBM



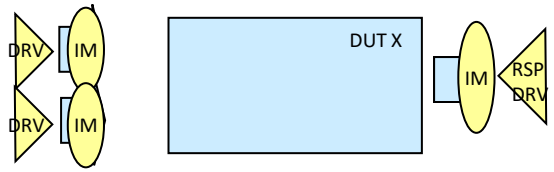




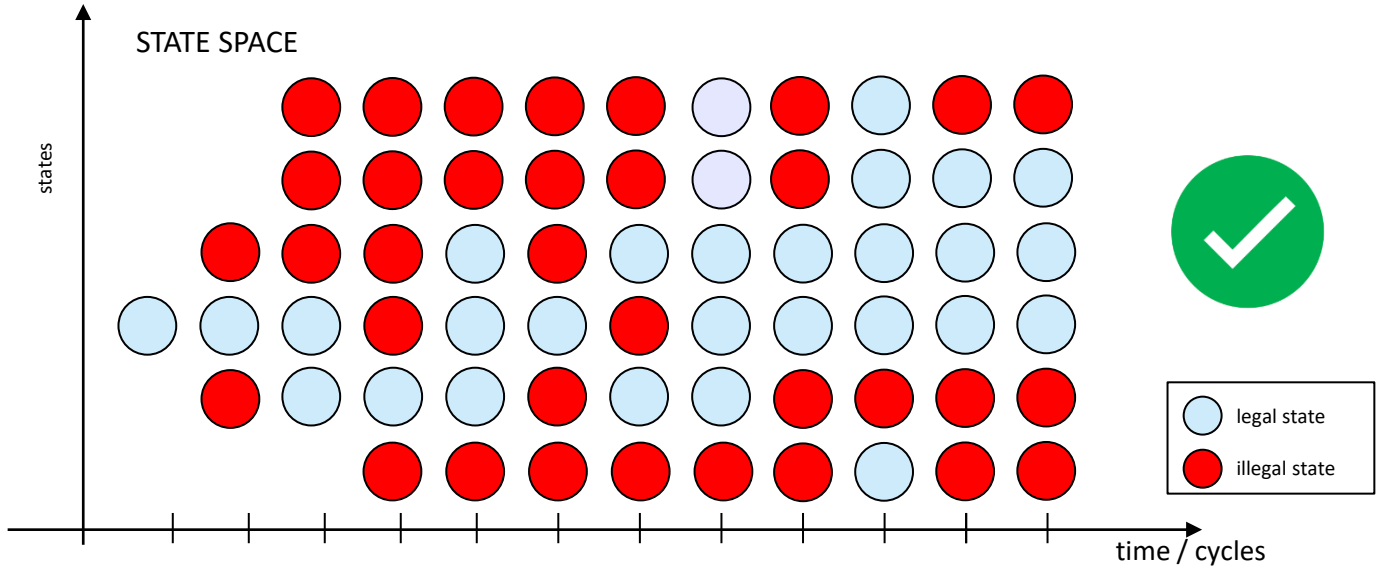
IBM View? The Power of Graph-Based Problem Solving



graphs are a great vehicle to describe the legal set of states through a combination of directed sequences (where state set is sparse) and random decisions (where state space opens up)

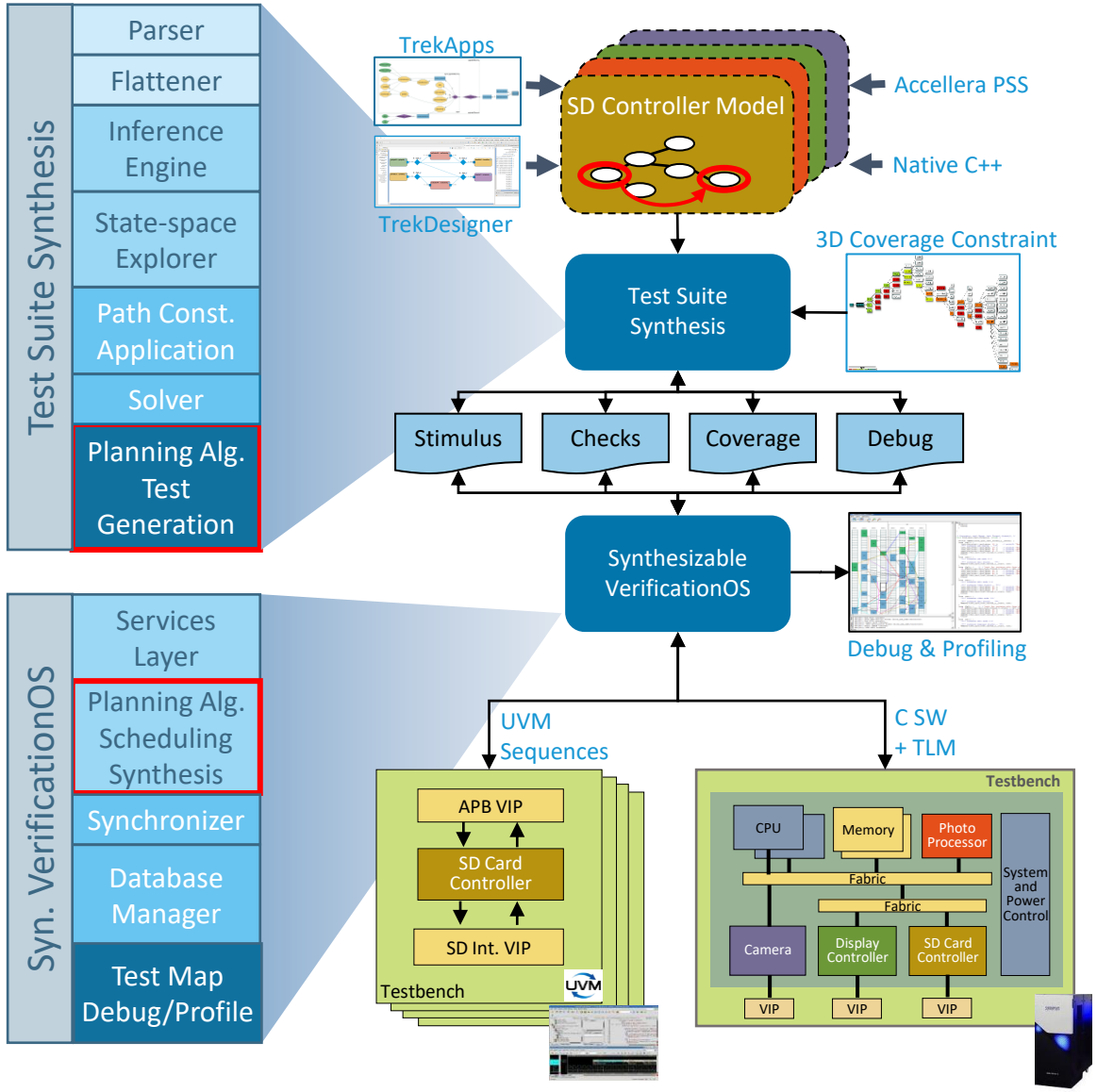


Courtesy



Breker Technology: Test Suite Synthesis & The VerificationOS

- How to create high-efficiency, high-coverage test content?
- Two Test Suite Synthesis components used AI-based Planning Algorithms
- Planning algorithms are at the heart of optimized test generation and test scheduling



AI-Based Planning Algorithms and Verification

- **What is AI Planning?**

  *Planning is a long-standing sub-area of Artificial Intelligence (AI). Planning is the task of finding a procedural course of action for a declaratively described system to reach its **goals** while **optimizing overall performance** measures. Automated planners find the transformations to apply in each given state out of the possible transformations for that state. In contrast to the classification problem, planners provide guarantees on the solution quality.*

  Courtesy: IBM Watson

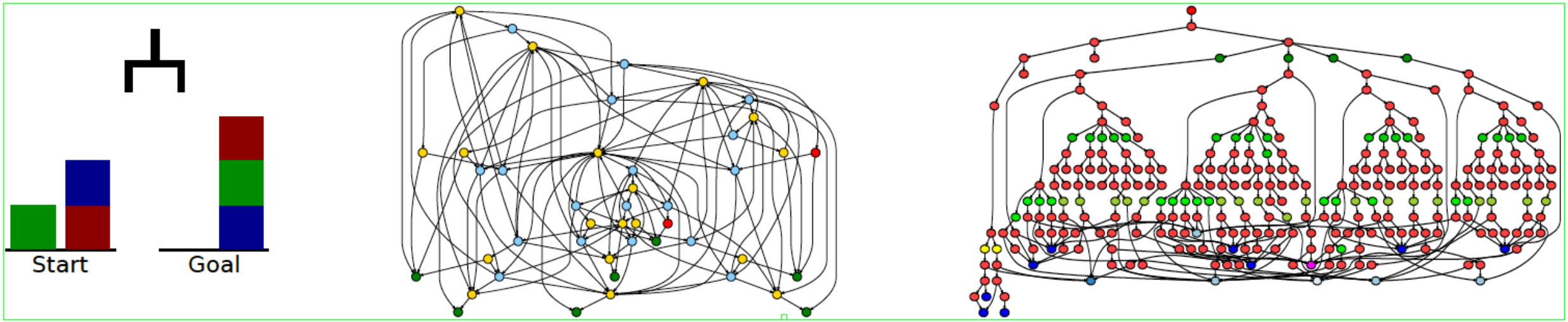
- Start with the desired outcome to check
- Solve for a plan of steps that will achieve that outcome

AI Planning Algorithm Theory

Online Planner Selection with Graph Neural Networks and Adaptive Scheduling

Ma, Ferber, Huo, Chen, Katz November 2019

IBM, University of Basel, Saarland University



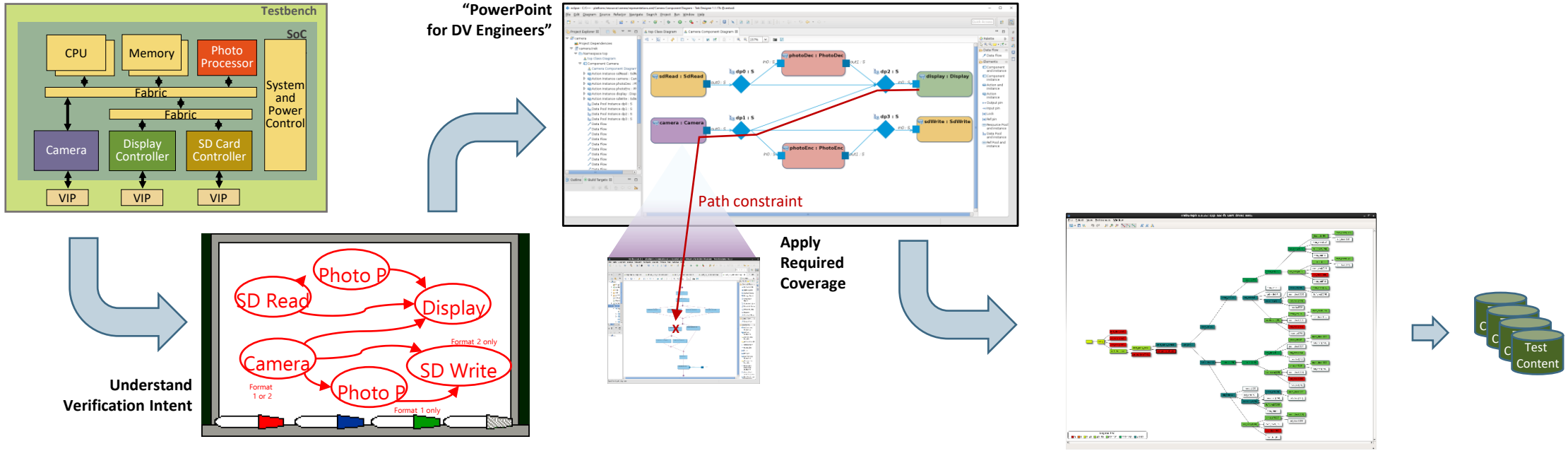
Gripper Example: Gripper needs to rearrange blocks.

- Graph describes a specification of all potential solutions.
- Flattened Graph shows all possible flows through specification.
- Planning Algorithm starts with the end in mind and works backwards to find optimum solutions

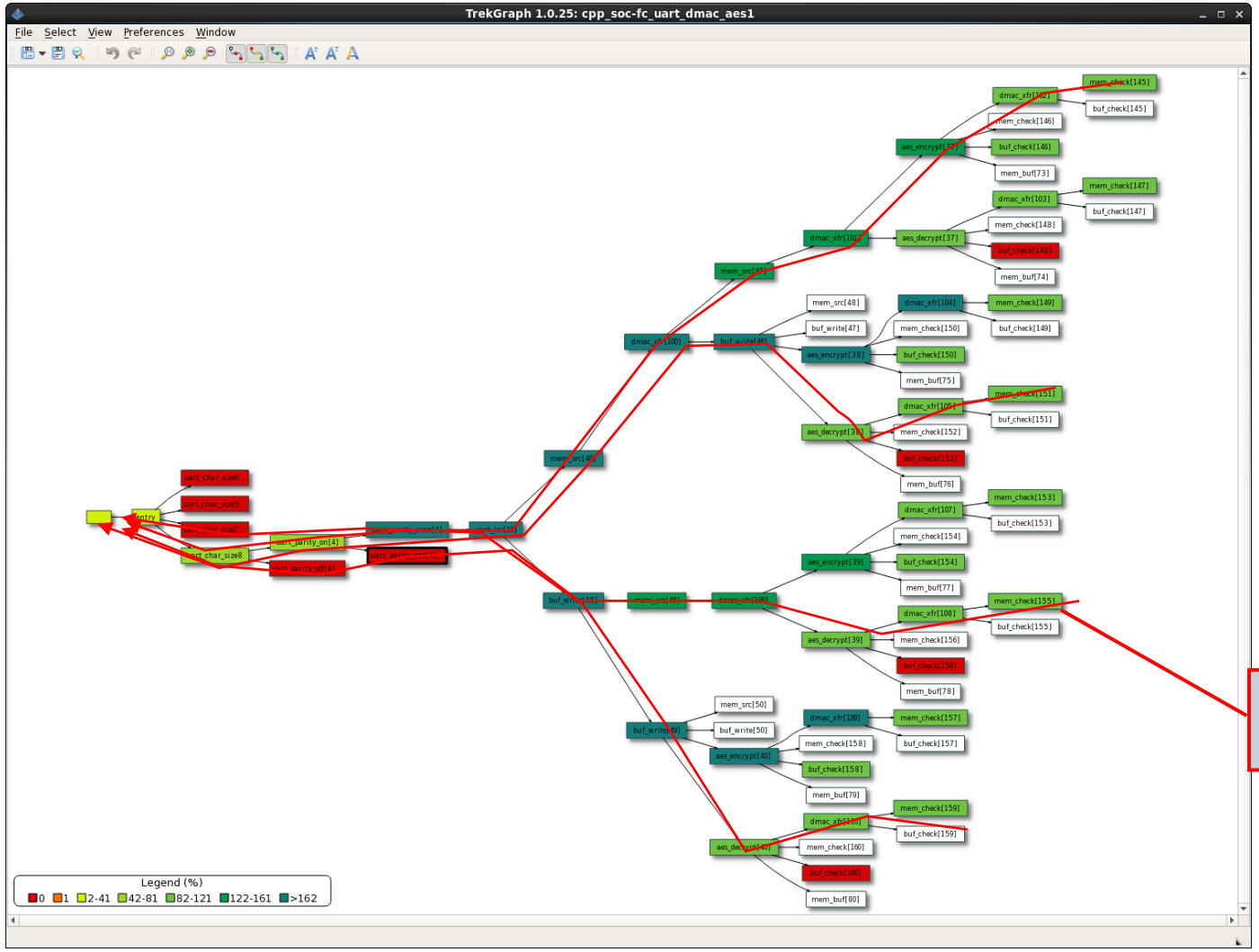
AI Planning Algorithms for Test Generation

AI Planning Solution

- Capture human-friendly path-constrained hierarchical flow charts
- AI Planning Algorithm infers coverage-optimized test suite



Sequential Reachability Analysis and Coverage Analysis

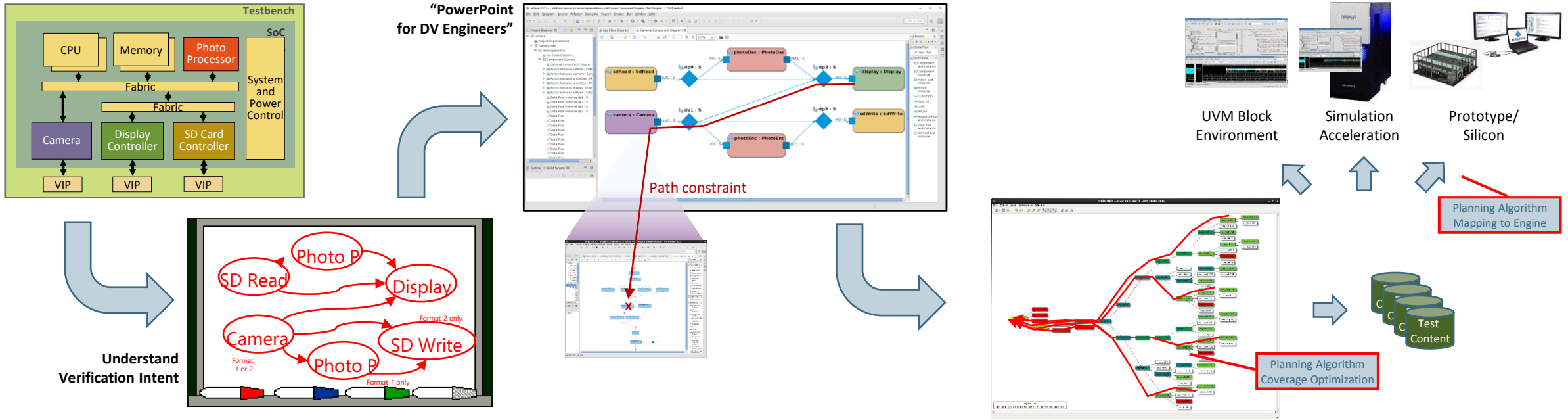


Planning Algorithm
Coverage Optimization

AI Planning Algorithms for Test Generation

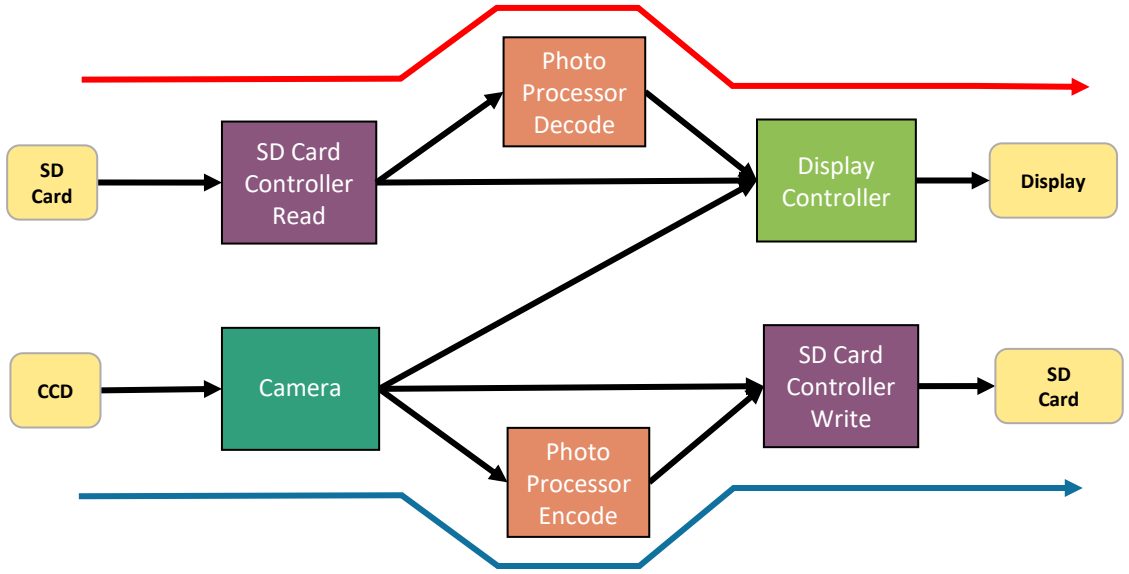
AI Planning Solution

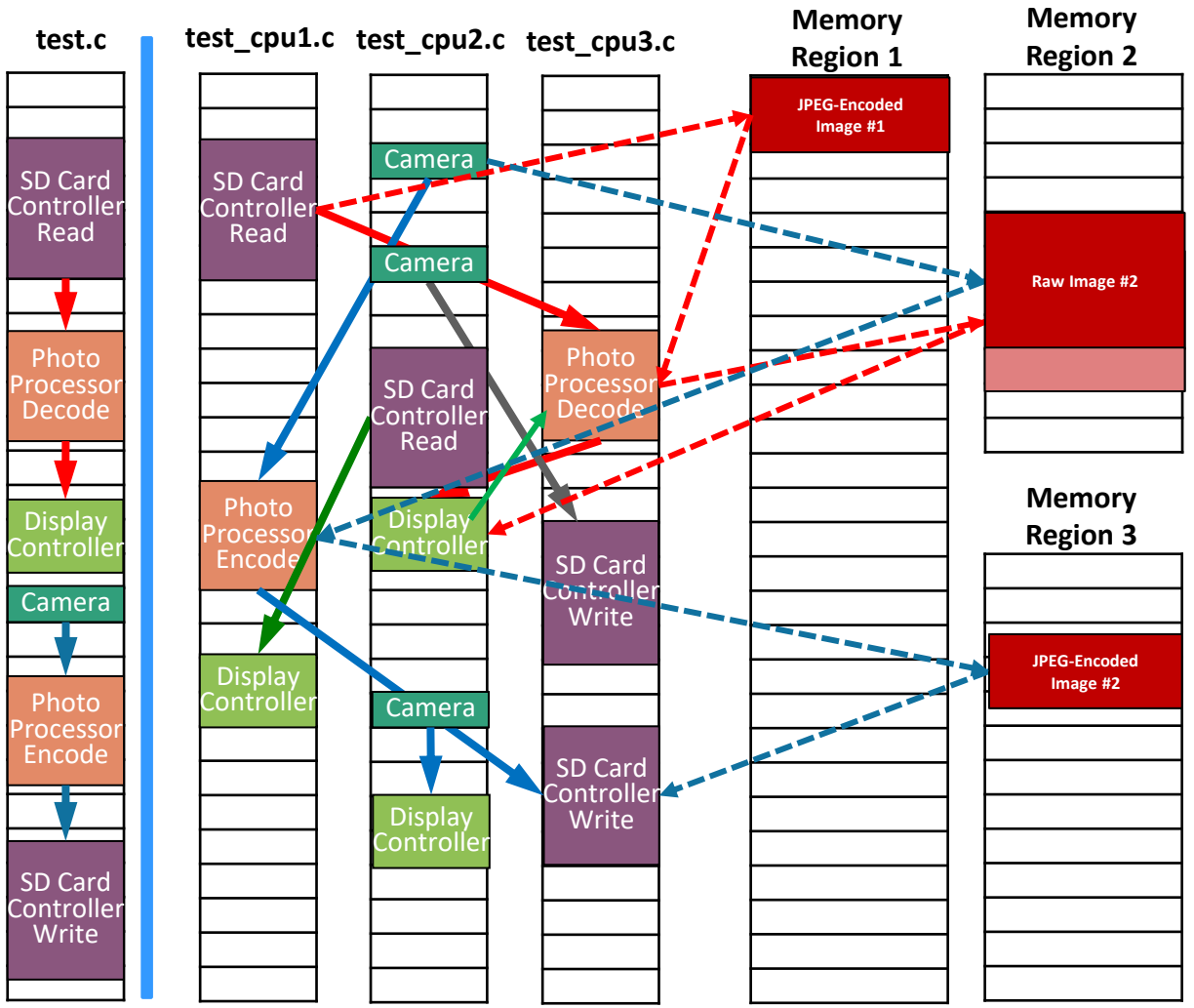
- Capture human-friendly path-constrained hierarchical flow charts
- AI Planning Algorithm infers coverage-optimized test suite
- AI Planning Algorithm infers mapping to execution engines



Tasks and Resource Scheduling

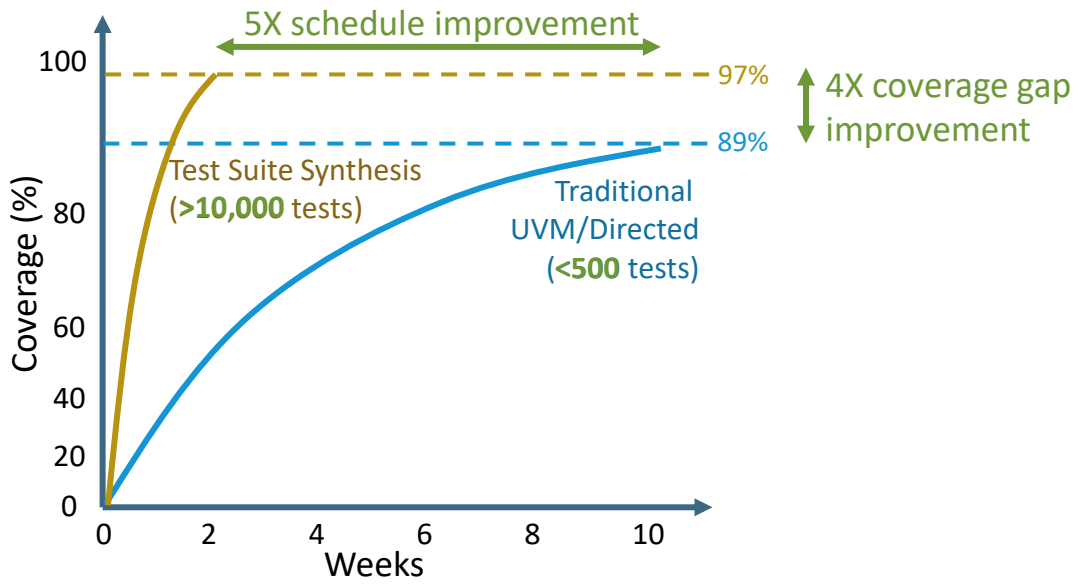
- Breker patented scheduling synthesis interleaves tests across resources
- Need to ensure resources are used correctly and possible bottlenecks are exercised
- Another opportunity for end in mind synthesis





Typical Customer Results

- Broadcom case study
- Complex cell phone SoC
- UVM/C tests augmented existing UVM testbench



Metric	Manual	Synthesis	Improvement
Test Authoring Time	2.5 months	2 weeks	5X
Unique, High-Impact Tests Generated	<500	>10,000	20X
Coverage Gap (100% - Coverage)	11%	3%	4X

Thanks for Listening!
Any Questions?