DESIGN AND VERIFICATION **CONFERENCE AND EXHIBITION**

UNITED STATES

SAN JOSE, CA, USA FEBRUARY 27-MARCH 2, 2023

Accelerate Coverage Closure from Day-1 with Al-driven Verification Malay Ganai, Will Chen

SYNOPSYS[®]

Srikanth Vadanaparthi

Qualcom



Agenda

- Overview
 - Intelligent Coverage Optimization (ICO) Technology
 - ICO Values: KPI, ROI
- Real Case Studies at Qualcomm
- Success Stories
- Q/A

SYSTEMS INITIATIVE

[25 min] [15 min] [10 min]

[40 min]





UNITED STATES

SAN JOSE, CA, USA FEBRUARY 27-MARCH 2, 2023

ICO Overview

Technology, Values

SYSTEM



Verification Tasks







Verification Asks

Reduce Number of Repeats \Rightarrow More Goals per Repeat = **Productivity** Gain

TB Dev, Debug



-25%

Functional Verification

Bug Fixes Cover Tests

2

Repeat

Coverage Holes RTL Bugs, TB Issues

More Goals Per Repeat

- TB Issues inefficiency, under/over constraints, missing scenarios, illegal stimuli
- DUT Bugs easy & hard-to-hits
- Cover group hits easy & hard-to-hit
- Code Coverage, Assertions, Cover Properties
- Spec/Checker issues, ...



Reduce Resource per Repeat \Rightarrow Less Cost per Repeat \equiv Efficiency Gain Less Cost Per Repeat

RTL

Sign Off

- Faster Performance
- Smaller Regression Size
- Fewer Grid Resource
- Easy deployment, Quality/Robustness
- Better debugging, Auto RCA
- Faster Bug Validation, ...



SYSTEMS INITIATIVE

~13%

Spec



ASK #1: Shift-Left Functional Coverage



TB Development Ask – Stabilize and Mature TB faster **Optimize Regression, Achieve Coverage Goals** Ask – Accelerate Coverage and Bug rate Coverage Convergence & Closure Ask – Faster closure, Save writing directed tests "ICO can help from Day 1"







Reduced efficiency (wasted cycles)

"ICO can help from Day 1"





Intelligent Coverage Optimization (ICO): Key Differentiators

2022.06, 2023.03

Enhances Stimulus Quality using Reinforcement Learning

- Exposes more bugs including TB latent issues
- Reduce regress TAT with boosted grading
- Faster and Better code and functional coverage

Provides TB Visibility & Analytics

- Stimuli distribution: diversity metric, histogram
- RCA to Low coverage, Skewed distribution, Under/Over-Constraints

Deployable at all stages of TB development

- Early (no user-coverage), Intermediate, Stable (rare issues)
- Ease of use and deployment (fits seamlessly in Synopsys VCS user environment)
- No need to rewrite/change functional coverage model







ICO KPIs







ICO: KPI \Rightarrow ROI

- Measuring ICO Values & Impacts
 - ICO KPI := Func(TB Constraint complexity, TB Randomness, TB Maturity)
 - When used in early/mid stages
 - Higher volume of bugs => faster TB stabilization [User1: weekly bug rate up 2X]
 - More coverage => write fewer directed tests, grid savings [User2: 64% after 7 vs 15 runs, speedup 2x]
 - TB insights => missing scenarios, incorrect constraints [User3: help revise TB to reach bins 100%, cross @ 99%]
 - When used in late stage
 - Hard-to-hit, latent, omission bugs => reduce repeat [User4: spec bug, deadlock]
- Cost of Results
 - Ease of Use, Low Overhead, Runtime On/Off flexibility, Product robustness => reduce cost





ICO: Blogs, Pubs, Presentations

HOME SYSTEMS &	& DESIGN LOW POWER -	HIGH PERFORMANCE	MANUFACTURING,	PACKAGIN
SPECIAL REPORTS	BUSINESS & STARTUPS	JOBS KNO	WLEDGE CENTER	TECHN
Accelerat With Intel Optimizat	ting Verific lligent Cov tion	ation Sh erage	ift Left	
New tools or techn expose bugs early i verification turnaro	ologies that can acce in the design cycle, re ound time.	lerate coverage co duce debug effor	onvergence, t, and improve	

Functional verification dominates semiconductor development, consuming the largest percentage of project time and resources. Team members look at the rate of design bug discovery, consider anecdotal information on the types of bugs that escaped to silicon in previous projects, and use their best judgment based on their years of experience to determine when to tape out. Above all, they look at various coverage metrics. Verification engineers want to see these metrics reach the target goals, or at least asymptotically converge toward these goals. Thus, any new tools or technologies that can accelerate coverage convergence, expose bugs early in the design cycle, reduce debug effort and improve verification turnaround time have high value by "shifting left" the verification process and therefore accelerate tape out and software development using fewer resources.







Many ICO Successful Deployment Stories from SNUG







ICO Value #1: Easy Use Model

No need to rewrite / change func model

No need for a merge server

Deployable on-prem / external cloud

<pre>% vcs //nothing needs to be changed % crg -dir <shared cdb=""> -shared init</shared></pre>	Initialize/Reuse Shared cdb
<pre>// run regression in parallel % simv <ico_run_opt> <other run_opts=""></other></ico_run_opt></pre>	Launch ICO Regression
<pre>% crg -dir <shared_cdb> -report crgReport</shared_cdb></pre>	Generate CRG reports









SYSTEMS INITIATIVE

ICO Value #2: Improve Stimuli Diversity







ICO Value #3: Boost (URG) Grading

		Improved Coverage			
Regression	Default	Default Grade	ICO	ICO Grade to Default Coverage	ICO Grade to Highest Coverage
Group Coverage	74.27%	74.27%	85.06%	76.35	85.06%
Tests Needed for Coverage Score	843	24	840	2	10
Run Time (CPU)	364 Hours	20.5 Hours	444 Hours	5 Hours	22.8 Hours
4.1x TAT reduction, Same Coverage					
73x TAT reduction, Same Coverage					
16x TAT reduction, Improved Coverage					





ICO Value #4: Accelerate Functional Coverage Rate



Early Mid Stage





ICO Value #5 & #6: Provide TB Insights, RCA

File Vjew Source Onefface Simulation Tools Window Help	🔎 cfg.device_delay_min 🛛 🗙 🔘 🗶
<mark>● 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</mark>	
CLASS VARIABLES Petted Covered lingui Excluded COVP	*Srcl:push_pull_agent_wush_pull_agent_cfg.sv) Line [11]
notic genery max 32 39 14 23 22 0 23/3 37.84 sck lo deby max 32 59 16 18 25 0 27.13 37.84 req lo deby max 32 59 16 18 25 0 27.12 47.06 req lo deby max 32 59 17 17 25 0 28.81 50.00 57.61 66.61 host, deby min 32 59 1 0 58 109.00 100.00 100.00	90 000100_00137 == 0,0 91 0 else 92 device_delsy_max dist (93 [1:10] // 1. 94 [11:50] :/ 4.
Bits Histogram VI AL VI	95 [51:100] :/ 3, 96 [10:500] :/ 2, 97 [501:1000] :/ 1 98); 99);
- UNCOVERD	100) 101 102 constraint req_lo_delsy_min_c (103 soft req_lo_delsy_min == 0, 104) 105
COVERED 3.95 COVERED 6.38 • COVERED 10.34 • COVERED 10.54	
Class infrance Class infrance erred line 2/src/lowine 0.000 0.000 0.000 0.000 0.000 2/src/lowine 0.000 0.000 0.000 0.000 0.000 0.000 2/src/lowine 0.000 0.000 0.000 0.000 0.000 0.000	HISTOGRAM
Jstrichmits, dr. y later, der, de gen, dem 12 311 62 0 249 Jstrichmits, dr. y lagen, Gu., (e) y losa, ten 12 349 123 0 26 0 34.38 100.00 40.74 94.12 Jstrichmits, dr. y lagen, Gu., (e) y losa, ten 12 349 123 0 26 0 34.38 100.00 40.74 94.12 Jstrichmits, dr. y lagen, Gu., (e) y losa, ten 12 349 123 0 26 0 34.38 100.00 39.66 91.08 Jstrichmits, dr. y lagen, Qu., (e) y lagen, ten 12 349 147 0 39.29 100.00 39.56 91.08 Jstrichmits, dr. y lagen, th, (e) y lagen, ten 12 349 147 0 39.29 100.00 39.56 95.09 Jstrichmits, dr. y lagen, th, (e) y lagen, ten 12 349 97 16 181 18.26 88.78 37.49 78.69 95.09	118 119 constraint ack_lo_delay_min_c (120 soft ack_lo_delay_min == 0; 121)
20% 20% 20% 30% 40% 50% 60% 70% 80% 90% 100%	122 constraint ack_lo_delay_max_c (124 solve zero_delays before ack_lo_delay_max;
	Highlights:
File Bule I'me Yiew Jools Window File Bule I'me Yiew Jools Window Stars of the Structure	•Bins: covered/uncovered/illegal
Reason for/src/lowrisc_dv_cip_lib_0/cip_base_env_cfg.sv:142:\push_pull_ RCA Analysis	•RCA capabilities •Potential over-constraints
<pre>6 rand bit[0:0] zero_delays; // rand_mode = ON 7 rand_bit[31:0] device_delay_max; // rand_mode = ON 9 constraint device_delay_max_c // (from this) (constraint_mode = ON) (/src/lowrisc_dv_push_pull_agent_0.1/push_pull_agent_cfg.sv:87)</pre>	•Triaging failures •Measure Progress
<pre>10 [1] zero_delays -> (device_delay_max == 0); 12 [:sero_delays) -> (device_delay_max dist {[1:10] :/ 1, [11:50] :/ 4, [51:100] :/ 3, [101:500] :/ 2, [501:1000] :/ 1});]3])</pre>	•Analyze Testbench Changes





ICO Value #7: Expose TB issues

• Found incorrect random distribution

std::randomize(num_zero_bytes) with { num_zero_bytes dist { [0:14] / 0, 15:= 1 }; }

• Changed many variables from int to bit vector to limit the boundary of legal values

```
int pkt_len; // Need to declare as bit[7:0]
std::randomize(pkt_len) with {pkt_len > 0; pkt_len <=256;};</pre>
```

• Found unwanted randomization of variables/constraint blocks

```
if(err_en == 1) begin // This condition was missing
   for(int k =0; k < 1000; k++) begin
      scan_frames[k] = new();
      scan_frames[k].randomize();</pre>
```

• Over constraint issues

```
constraint abc {
    abc inside {[0:2**n-1]} // Should be 0 : 2**n
}
```



Early Stage



UNITED STATES

SAN JOSE, CA, USA FEBRUARY 27-MARCH 2, 2023

Real Case Studies at Qualcomm

ICO values at Early, Mid, Late Stage of Projects

.

"Accelerating Functional Verification Through Stabilization of Testbench Using AI/ML" – DVCON'23

(



Requirements

- GPU architecture is intricate
 - Parallelism, complex logic
 - Challenging to find corner case bug, to achieve full functional coverage
- Manual effort is required for test creation
 - Semi-directed test cases to hit corner cases at the block level and cluster level
- Need for a productive approach with automation that
 - Does not require rewrite/change in functional coverage model
 - Helps shift-level verification accelerate bug finding, coverage rate
 - Ideally, fits seamless in our regression environment





Overview of Case Studies

- Demonstrate how we leveraged ICO
 - Deployed ICO in multiple real-life scenarios across multiple blocks
- Collected, analyzed data over a period of 2+ years (2019-2022)
- We present our observation across 3 projects at different stages
 - Project A @ Late Stage
 - Project B @ Early-Mid Stage
 - Project C @ Mid Stage
- Compare ICO vs. Non-ICO (status quo)
 - Key Performance Indicators: Bug rate, Coverage rate
 - ROI: productivity, efficiency





ICO Value #8: Cover Hard-to-Hit Targets

• The L2 cache TB was stable

SYSTEMS INITIATIVE

- pass rate of 98% with a coverage rate of 95% over 10 regression cycles.
- Full insight into stimulus types with ICO
 - Uncovered 30% more bugs despite TB being stable.



Improved testbench, expose rare scenarios

- <u>30% more TB Issues uncovered:</u> Constraint inconsistency failures, Design SVA failures due to issue in TB UVM driver, Scoreboard issues related to checker
- o RTL Issue uncovered: Design deadlock scenarios
- o Full visibility into TB (over-constrained, distributions, RCA)
- o 6 regressions with ICO vs. 10 regressions with Default to hit 95% functional coverage (each regression takes 1 day, 4 days of savings)

	Default simulation UVM error count per regression	ICO simulation UVM error count per regression
Regression 1	67	84
Regression 2	64	86
Regression 3	70	79
Regression 4	72	111
TOTAL	TOTAL=273	TOTAL=360
Inconsistent constraints: 0 Inconsistent constraints: 89# Scoreboard mismatch errors: 0 Scoreboard mismatch errors: 7*		





Project B: Early Mid Stage

ICO Value #9: Shift-left Bug Rate

More Goals Per Repeat => Higher Productivity



- ICO was enabled on Week 6th when pass rate was ~98%
- A big surge in both TB and DUT was observed even when TB/DUT was deemed stable
- In subsequent fewer iterations all DUT bugs were exposed





ICO Value #10: Save Writing Directed Tests, Grid Usage

Runtime by Block over time

Courrage	Matria	Default	ICO Enchlad	Improvement
Coverage	Wietife	Default	Ellabled	
	Line	88064	89231	1167
Cala	FSM	855	885	30
Code	Condition	356836	359019	2183
	Branch	66888	69931	3043
	Cover Points	1757	1804	47
Functional	Coverpoint bins	39628	39969	341
Functional	Cross Coverage bins	305445	306859	1414
	Total Bins	347984	349661	1677

Render Block has ~350K coverage bins

- Combinations of color/depth formats, tile packings
- Required months of back-to-back regressions



Default: 198k time units



10-15% reduction in grid usage per block (Project B)

"After enabling ICO in random regression for this block, we witnessed dramatic improvement or left shift in functional coverage closure by 1.5 weeks. We observed that most of complex bins which needs tweaking of constraints or delay profile automatically are now getting hit quickly. Consequently, we now need to spend very minimal effort (maybe for less than 100) bins instead of 200 to 500 bins) on directed /constraint tweaking or delay profile tweaking.





Project C:

Mid Stage



UNITED STATES

SAN JOSE, CA, USA FEBRUARY 27-MARCH 2, 2023

ICO Success Stories

ICO values at various design houses



ICO Success Stories



RESULTS

- Exposed 51 Testbench bugs
- 10% increased coverage with hard to hit bugs exposed



RESULTS

- >10% coverage improvement
- Exposed rare scenarios
- Production use in IP/SoC blocks



RESULTS

- Upto 3X faster TAT
- Regression Count reduced by 2X
- Exposed DUT bugs





Early and Additional Bugs Found on a Mobile Design

1.67X faster coverage closure and 30% more issues uncovered

Application	Graphics	
Customer Goals	Results with VCS ICO	
Insight into stimulus coverage	 Full visibility into testbench (over- constrained, distributions, RCA) 	
Shift left in bug hunting	 30% more TB Issues uncovered RTL issue uncovered: design deadlock scenarios 	
Verification completeness	 1.67X faster coverage closure 6 regressions with ICO vs.10 regressions with default to hit 95% functional coverage 	







Faster and Improved Coverage at a 5G Design

1.5X faster coverage closure, 5% higher coverage compared to default regression

Application	5G, IoT	5G, IoT, mission critical systems			
Customer Goals	5	Results with VCS ICO			
Testbench visibility ar find more bugs	nd • Visibil divers • Expos	ity to stimulus covera ity and distribution sed one TB issue	to stimulus coverage, and distribution one TB issue		
Increase coverage and faster coverage	• 5% hi reach	• 5% higher coverage and 1.5X faster to reach complete coverage closure			
Regression	Default	VCS ICO			
Group Coverage	99.81%	99.81%			
Tests Needed for Coverage Score	911	507			
Run Time (CPU)	36.76 Hours	1.5x 23.79 Hours			

SYSTEMS INITIATIVE







Improved Quality and Compute Efficiency on a CPU Design

Exposed 2X failure buckets, 4% higher coverage with <30% runtime overhead

Application	Microprocessor	
Customer Goals	Results with VCS ICO	
Improve verification quality	 Found 2X buckets of failures with the same seeds Reports provide interesting insights on efficacy of stimulus 	Toil
Increase Compute Efficiency	<30% runtime overheadComparable cycle per second	Run
Increase Verification Efficiency	 Cover Group score per regression seems to be 4% higher on average 	Cov Stim



Metric	Default	VCS ICO
Failure Buckets	17	32
Run time (CPS)	67 Hz	69Hz
Coverage	71%	75%
Stimulus Quality	Cov: 9.44% Diversity: 2.74%	Cov: 26.75% Diversity: 12.78%





Summary: ICO Helps Shift-Left Verification From Day-1



Verification Productivity



Verification Efficiency

Project Timeline





2023 DESIGN AND VERIFICATION TO DVCDN CONFERENCE AND EXHIBITION

SAN JOSE, CA, USA FEBRUARY 27-MARCH 2, 2023

Q&A





UNITED STATES

SAN JOSE, CA, USA FEBRUARY 27-MARCH 2, 2023

Thank You

Contact Synopsys AE's for More Information

vcs_support@synopsys.com

SYSTEMS INITIATIVE

