Coverage Closure – Is it a “Game of Dice” or “Top 10 Tests” or “Automated Closure”?

Aditya Sharma, T. Nagasundaram,
M. Nitin Kumar, Nikhita Raj J

accellera SYSTEMS INITIATIVE™

VerifLabs

2015 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Agenda

- Conventional Coverage Closure : CRV
- Smart Selection of Coverage Rich Tests : Test Ranking
- Automated Coverage Closure : GBV
- Summary & Conclusion

© Accellera Systems Initiative

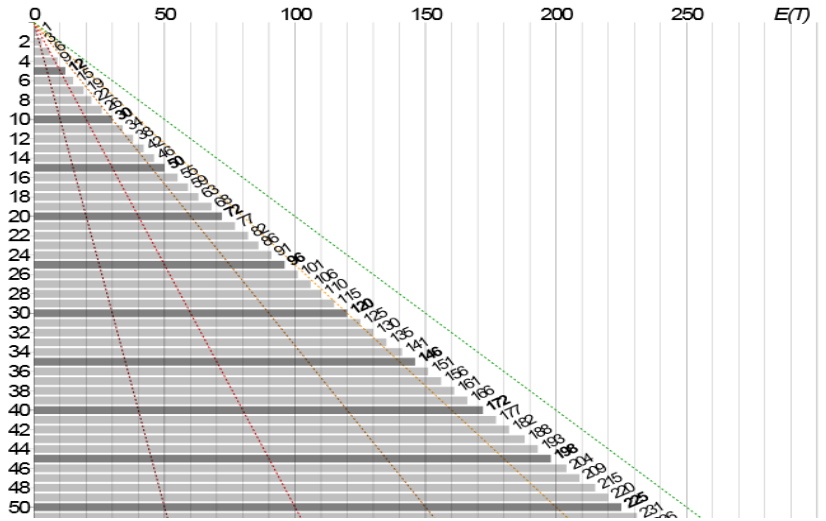
Conventional Coverage Closure

- "Calling-it-a-day" in a typical SoC verification project?
  - approach is to use the coverage metrics available
- Though not silver bullets (functional & assertion coverage), but are the most formidable
- Now with the acknowledged coverage metrics, the process involves three phases:
  - Coverage definition
  - Coverage collection (running simulations)
  - Coverage analysis & closure (iterative)

© Accellera Systems Initiative

CRV – more of a “Game of Dice”

- Hit & Try
- Typical constraint random generator has well known redundancy built in
- Coupon Collector’s Problem
- Expected number of trails grows as Θ(nlog(n))



© Accellera Systems Initiative

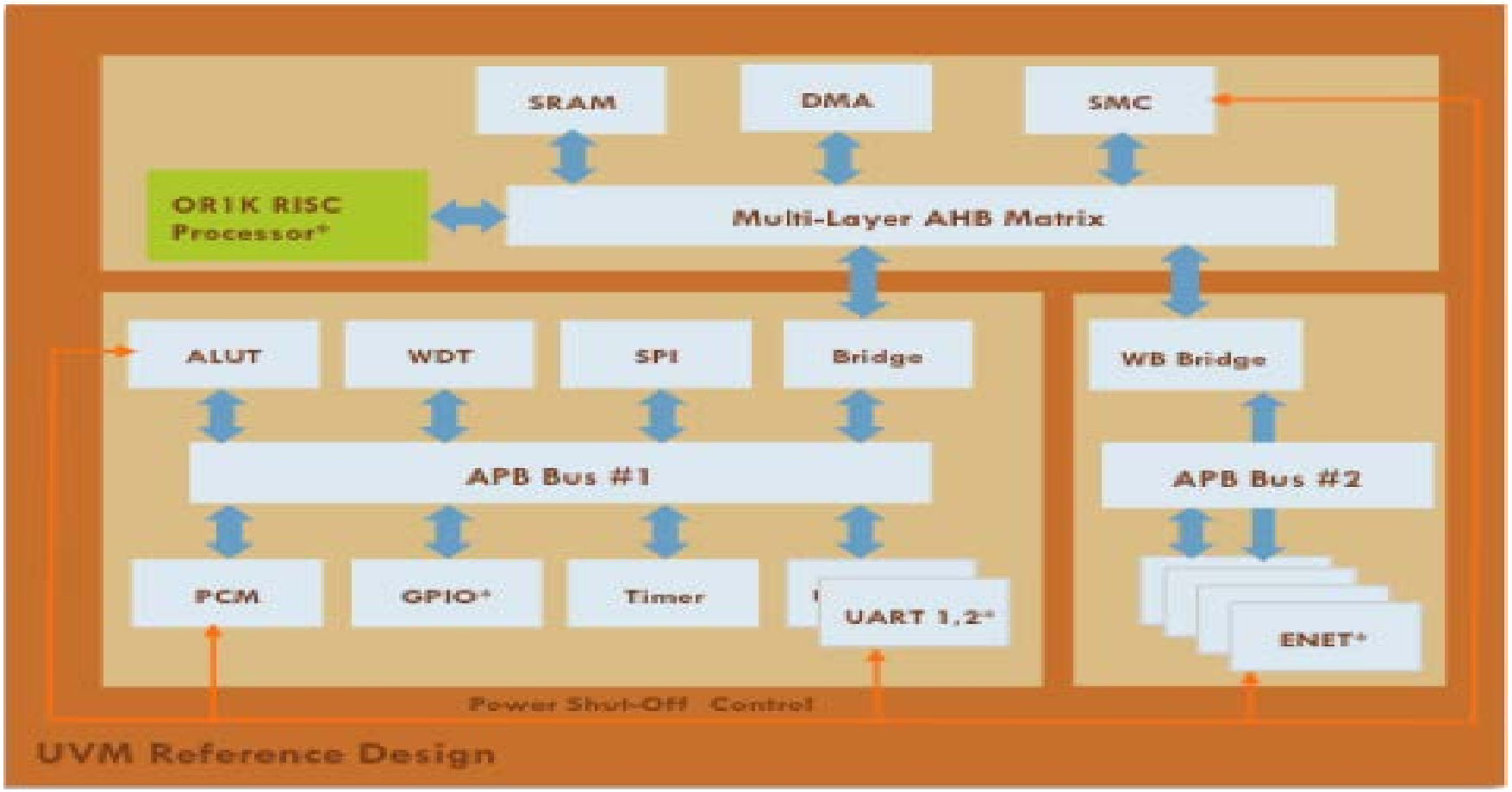
Game of Dice - Samples

- **APB-SPI IP**
  - Number of Coverage Points or (State Space) : 192 (N)
  - NUM_XACTNS needed for 100% Coverage ≥ N*ln(N)
  - N*ln(N) : 1009

- **UVM SoC Reference Kit**
  - Number of Coverage Points or (State Space) : 2749 (N)
  - NUM_XACTNS needed for 100% Coverage ≥ N*ln(N)
  - N*ln(N) : 21769

© Accellera Systems Initiative

A typical UVM environment for a SoC looks as below:



© Accellera Systems Initiative

Smart selection of Coverage Rich Tests : Test Ranking

- SoC verification comprises of several tests for single & multiple peripherals
- Typical regression of these individual tests are run with different seeds to cover a good state-space
- Now the question is how to arrive at a minimal sub-set to cover reasonable state-space automatically?

© Accellera Systems Initiative

Test Ranking

- This is where a Test Ranking feature can be very useful!

- The challenge is : Out of existing huge set of tests (and seeds) identify the top contributors!

Test ranking in UVM

- Most EDA tools support this feature today
- We have used Aldec's Riviera-PRO
- Below are few screenshots from other vendor tools (for completion sake)

VCS

Total Coverage Summary

SCORE	GROUP
39.79	39.79

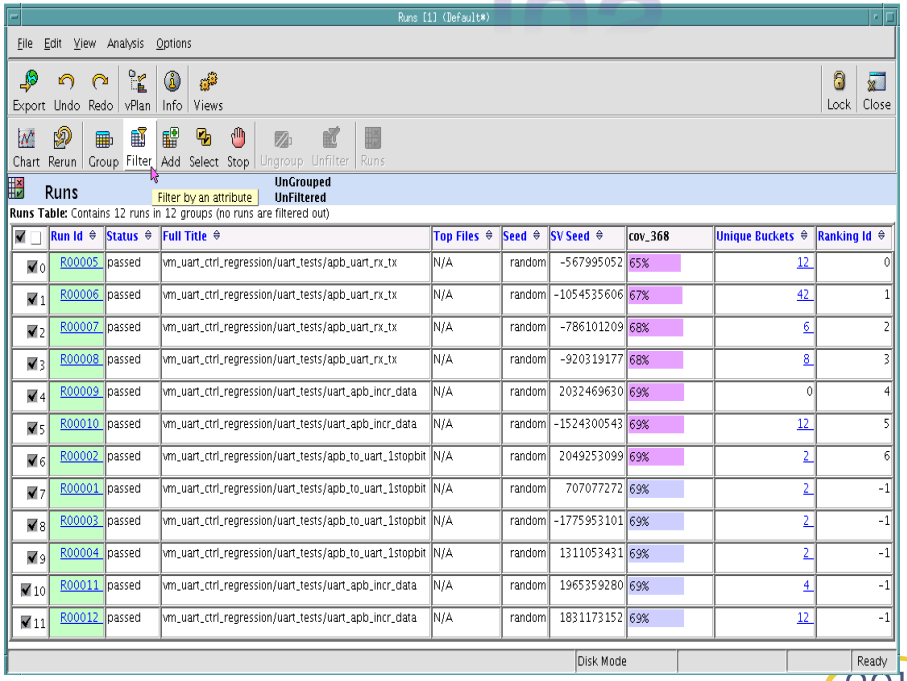
Total tests in report: 136

Tests are in graded order

Shows the accumulated (Total), standalone (Test), and incremental (Incr) score

SCORE			GROUP			
TOTAL	TEST	INCR	TOTAL	TEST	INCR	NAME
22.82	45.64	22.82	22.82	45.64	22.82	/remote/in01dws002/mohanss/r
33.27	45.53	10.45	33.27	45.53	10.45	/remote/in01dws002/mohanss/r
37.60	29.71	4.33	37.60	29.71	4.33	/remote/in01dws002/mohanss/r
38.60	29.71	1.00	38.60	29.71	1.00	/remote/in01dws002/mohanss/r
39.60	29.71	1.00	39.60	29.71	1.00	/remote/in01dws002/mohanss/r
39.73	32.26	0.13	39.73	32.26	0.13	/remote/in01dws002/mohanss/r
39.76	33.69	0.03	39.76	33.69	0.03	/remote/in01dws002/mohanss/r
39.79	34.00	0.03	39.79	34.00	0.03	/remote/in01dws002/mohanss/r
39.79	8.33	0.00	39.79	8.33	0.00	/remote/in01dws002/mohanss/r

IUS



© Accellera Systems Initiative

Test Ranking Results : UVM SoC Reference Kit

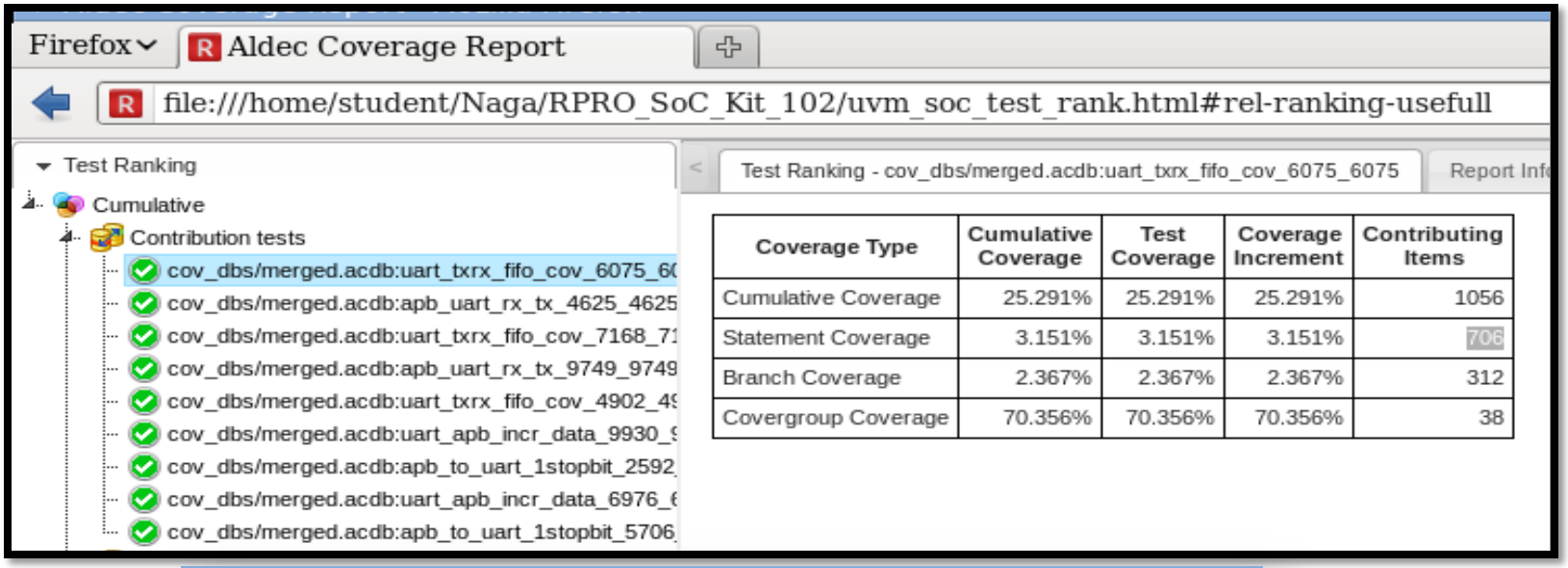
- UVM SoC Kit – Initial donation by Cadence
- CVC ported to other tools, scripts, updated code
- Freely available at www.accellera.org

TCL commands

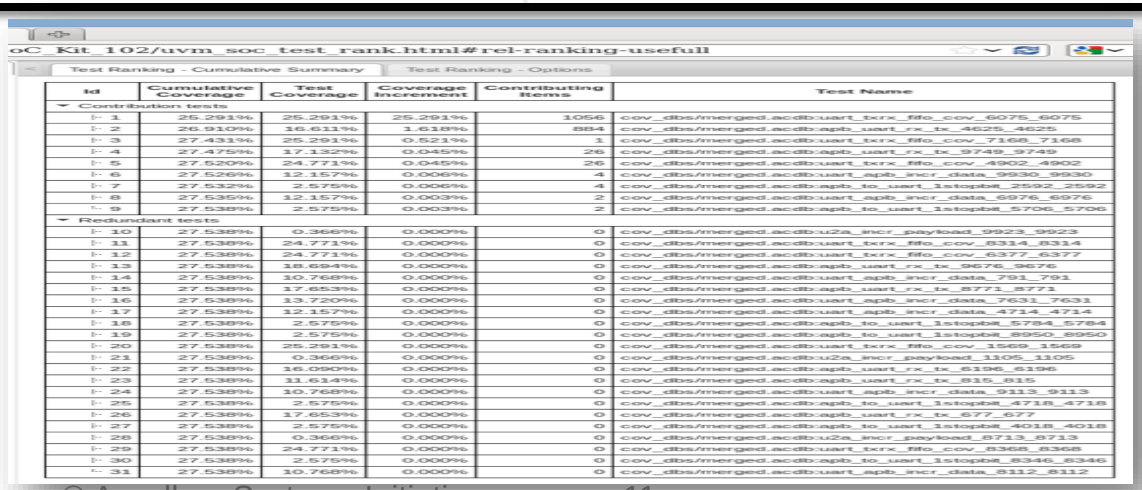
```
<for each test>
  acdb save -file cov_dbs/$TEST_SEED_NAME.acdb -testname $TEST_SEED_NAME

# Merge coverage results
acdb merge -associative -i cov_dbs/*.acdb -o cov_dbs/uvm_soc_merged.acdb

# Rank the tests as per coverage metrics

acdb rank -i cov_dbs/uvm_soc_merged.acdb -html -o uvm_soc_test_rank.html
```


© Accellera Systems Initiative

Test Ranking Results with Aldec

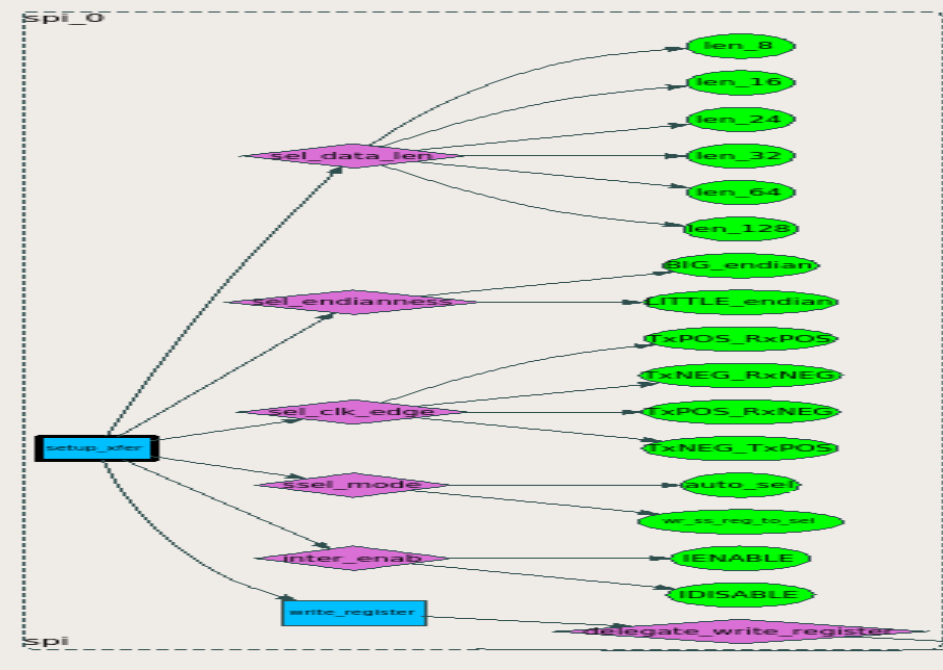


Coverage Type	Cumulative Coverage	Test Coverage	Coverage Increment	Contributing Items
Cumulative Coverage	25.291%	25.291%	25.291%	1056
Statement Coverage	3.151%	3.151%	3.151%	706
Branch Coverage	2.367%	2.367%	2.367%	312
Covergroup Coverage	70.356%	70.356%	70.356%	38



Automated Coverage Closure : GBV

- GBV enables capturing entire state space one-shot in a single graph
- N vs. N*ln(N) advantage
- Stimulus coverage goals often correlate to existing functional coverage goals
- Hence facilitating faster coverage closure



APB-SPI Use Case : Cover Group

Bit #	31:15	14	13	12	11	10	9	8	7	6:0
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Name	Reserved	INTR_STATUS	ASS	IE	LSB	Tx_NEG	Rx_NEG	GO_BSY	Reserved	CHAR_LEN

```
covergroup ctrl_reg_cvg;

   ssel_mode      : coverpoint trans.pwdata[13] ;
   inter_enb      : coverpoint trans.pwdata[12] ;
   sel_endinness  : coverpoint trans.pwdata[11] ;
   sel_clk_edge   : coverpoint trans.pwdata[10:9];
   sel_data_len   : coverpoint trans.pwdata[6:0] { bins len_8   = {7'h08};
                                                   bins len_16  = {7'h10};
                                                   bins len_24  = {7'h18};
                                                   bins len_32  = {7'h20};
                                                   bins len_64  = {7'h40};
                                                   bins len_128 = {7'h0};
                                                 }

   cross_cov: cross ssel_mode,inter_enb,sel_endinness,sel_clk_edge,sel_data_len;

   option.per_instance = 1;
   option.name = "MyCG";

endgroup : ctrl_reg_cvg
```

- Closer look to covergroup depicts total 192 cross coverage bins

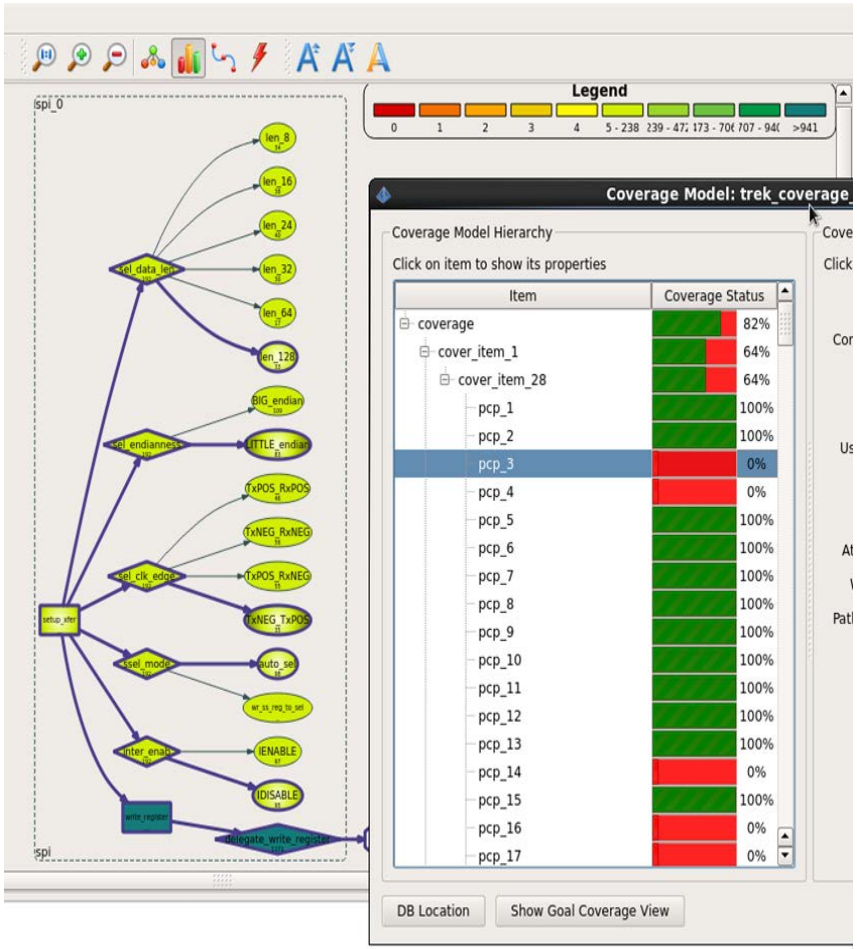
© Accellera Systems Initiative

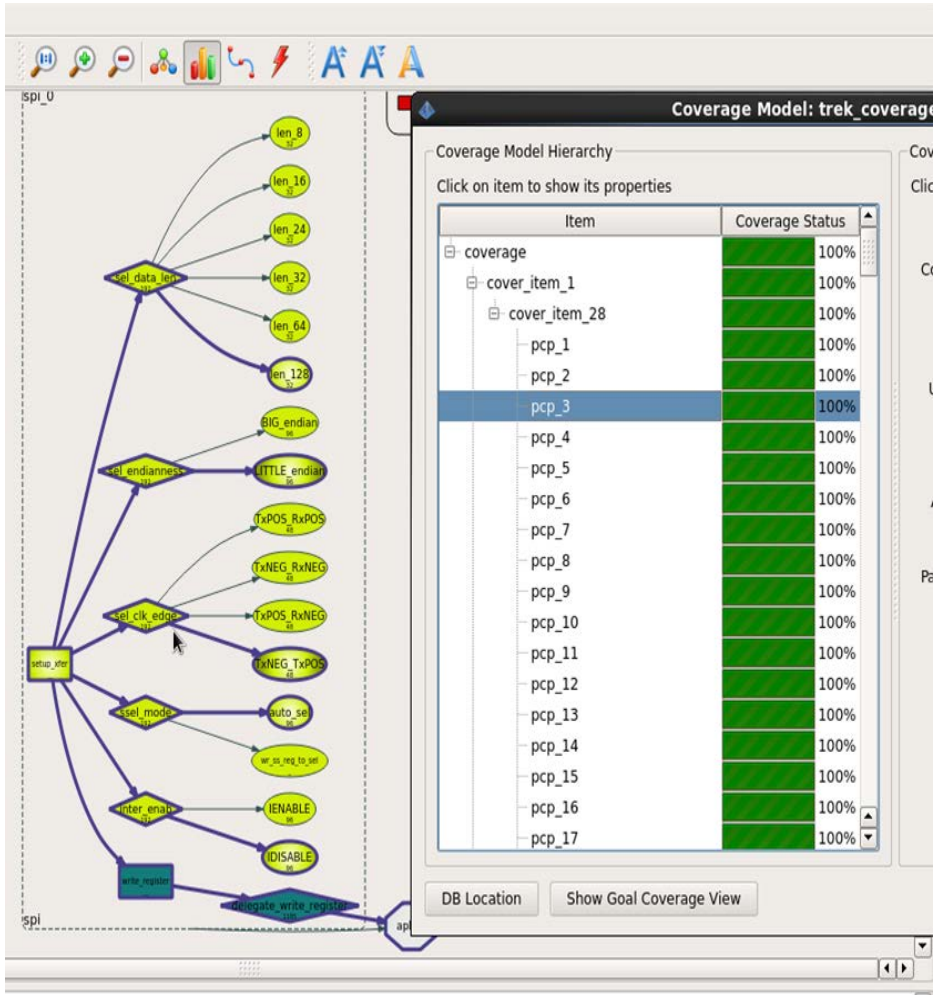
GBV with TrekUVM

- In this work we used Breker Verification System's TrekUVM as a test generator & APB-SPI UVM verification environment as a case study
- Automated Coverage Closure can be a reality with a model of state space driving a test generator to provide the needed UVM sequences
- 192 paths corresponding to 192 cross coverage bins for APB-SPI can be depicted in a single graph
- Turn On Coverage Closure Mode
- Automatically fill the holes & get 100% Coverage ☺

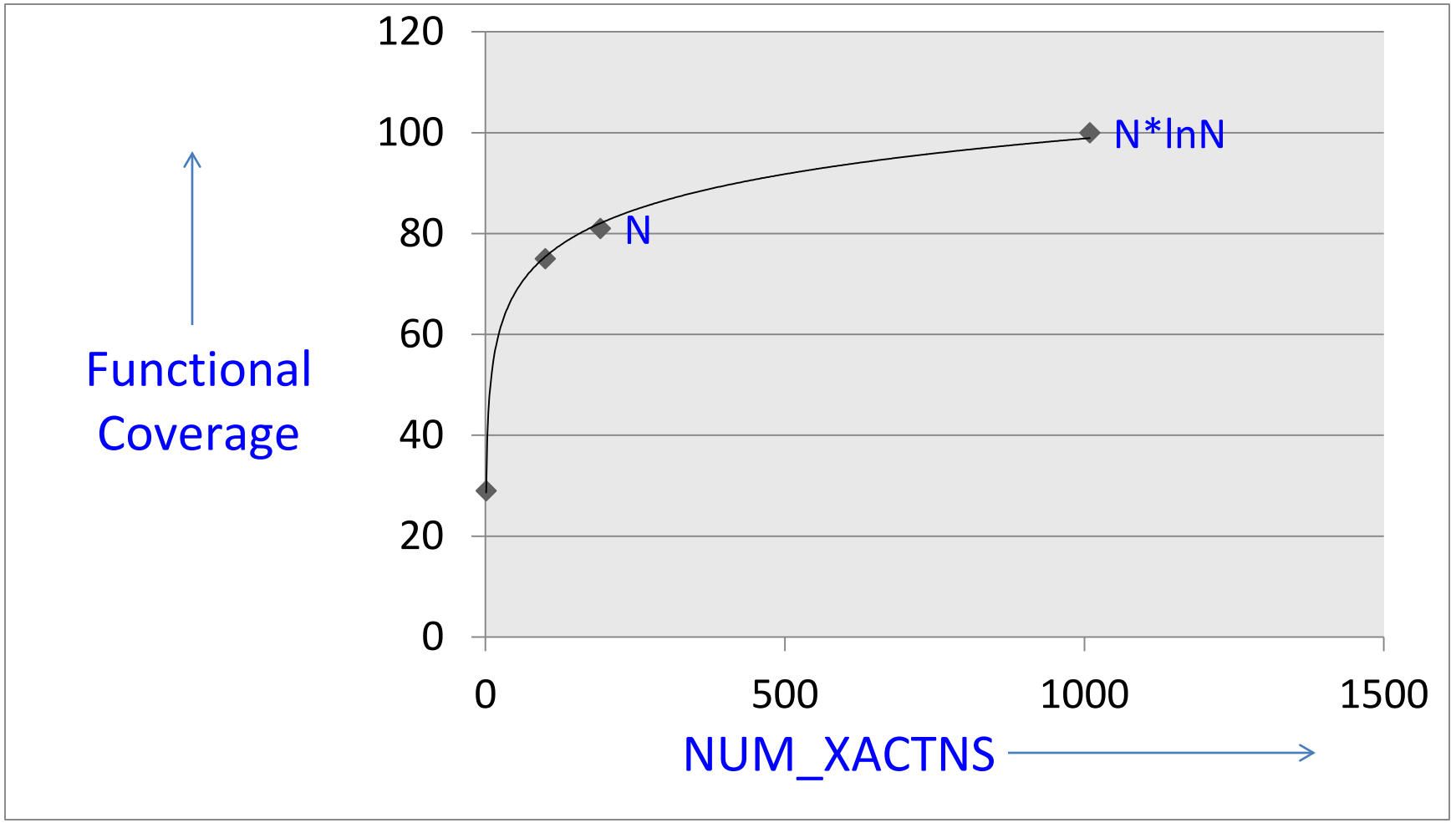
© Accellera Systems Initiative

Auto Coverage Mode OFF/ON





NUM_XACTNS vs. FCOV : APB-SPI IP



© Accellera Systems Initiative

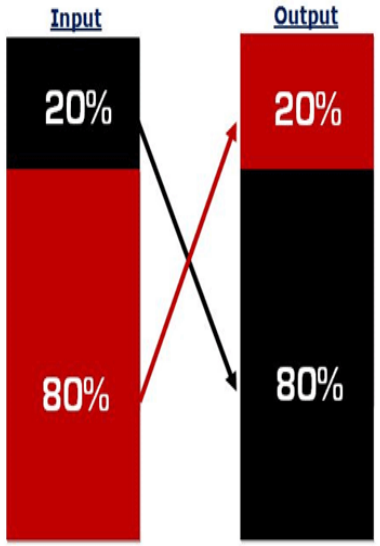
Summary

- **CRV** : Redundancy is inherent. Good at times, many a times too much redundancy. Leads to lot of manual coverage closure
- **Test Ranking** : Helps identifying high value tests. No rescue from manual coverage closure
- **GBV :** Eliminates redundancy totally while still preserving constraint randomness. Small effort to capture state-space as graph, fills coverage holes automatically.
  - We used it in this case study for input stimulus only

© Accellera Systems Initiative

Conclusion

- CRV is closely correlated to Coupon Collector Problem in probability theory
- First 80% vs. Rest 20% - 80-20 rule
- Coverage Closure above 80% solely via randomness is not feasible ☹
- Lesser time to tapeout-s
- Coverage closure via smart automation techniques can become the key differentiator technology



© Accellera Systems Initiative

Reference

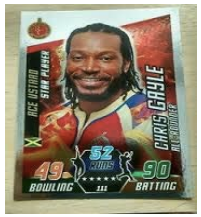
- *Coupon collector's problem: http://en.wikipedia.org/wiki/Coupon_collector's_problem*
- *Model Based Testing: https://en.wikipedia.org/wiki/Model-based_testing*
- *Breker Verification Systems (http://www.brekersystems.com)*
- *http://tinyurl.com/uvmsoc*

© Accellera Systems Initiative

Questions…?

The Coupon Collector's Problem

- N – total number of different coupons.
- X_i – time to get the i-th coupon.

$$E\left[\sum_{i=1}^{N} X_i\right] = \frac{N}{N} + \frac{N}{N-1} + \cdots \frac{N}{1} = N\left(\ln N + c + \Theta\left(\frac{1}{N}\right)\right)$$

- Only one coupon in a pack.
- Target is to collect all coupons.
- How many packs to be bought?