Using OneSpin Quantify

Darren Galpin, Infineon UK

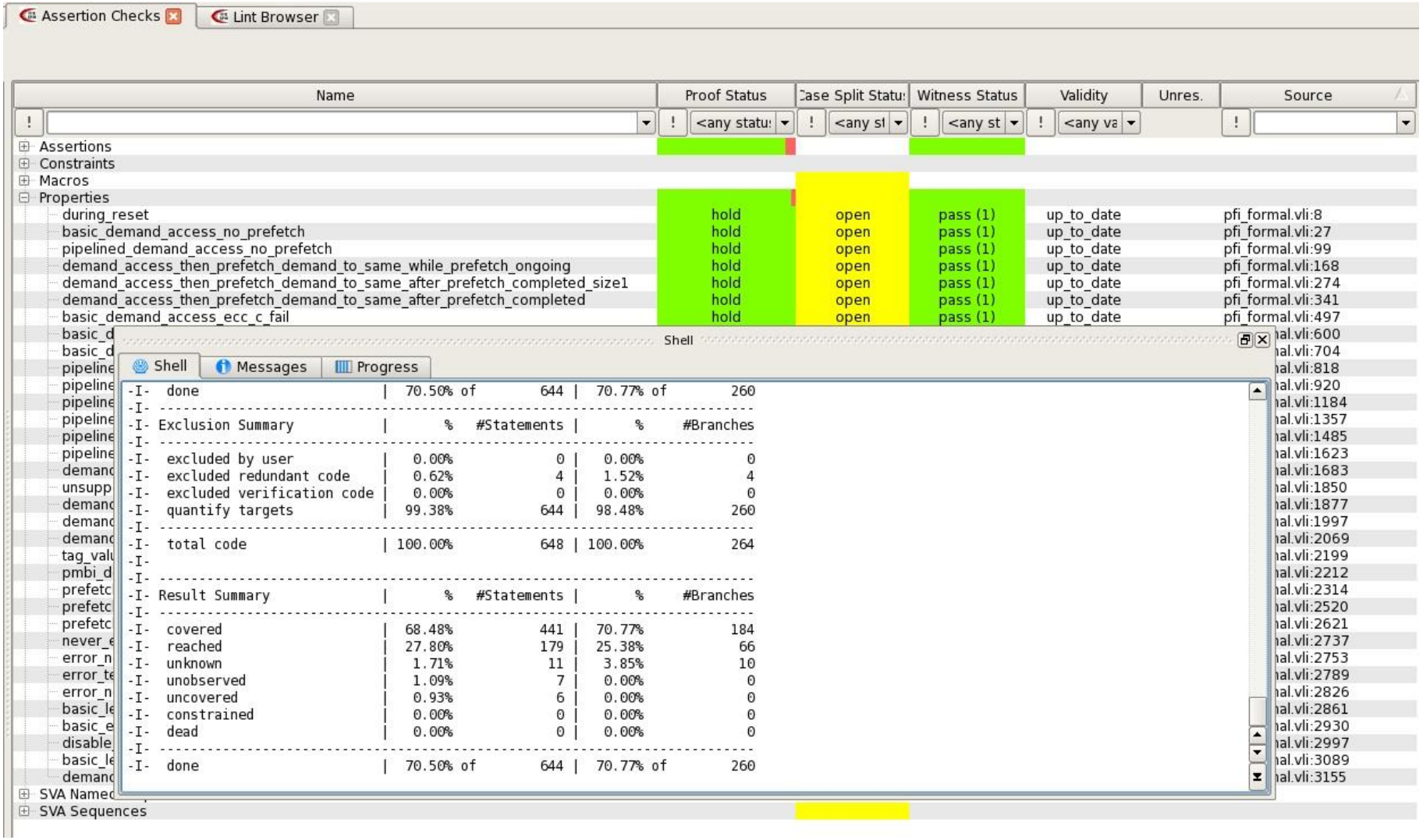
OneSpin Quantify – what is it?

- A tool for gauging the quality of property/assertion sets using all available assertions/properties.
- Generates a structural coverage metric for the code under test.
- Use to identify holes in the formal environment which completeness checks might have missed, or might have been missed due to constraints.
- Uses SVA, ITL, PSL and VHDL assertions together.

How to run

- Prove all properties/assertions once.
- Type “quantify” on the command line.
- Wait.....
- Analyse results.

Quantify running



What the summary means

- Only runs on executable design code – ignores code it thinks is redundant, or is included in embedded design assertions.
- Covered – The line of code was executed, and the effect of the line was observed. Similar but not identical to a Certitude fault insert and detection.
- Reached – Line of code was executed, but observation of effect not known. Equivalent of simulation statement coverage.
- Unknown – Haven't executed line, don't know if it can be observed.
- Unobserved – Line of code was executed, and proven not to be observable.
- Uncovered – Haven't executed line, proven not to be observable.
- Constrained – Can't hit due to constraints.

On finishing run.....

- Once run has been completed, generate one of two reports.
- 1) HTML.....

Structural Coverage Overview

Status		Statements		Branches	
1	covered	10	52.63%	7	58.33%
R	reached	0	0.00%	0	0.00%
U	unknown	0	0.00%	0	0.00%
0R	unobserved	2	10.53%	1	8.33%
0	uncovered	5	26.32%	2	16.67%
0C	constrained	1	5.26%	1	8.33%
0D	dead	1	5.26%	1	8.33%
Sum	quantify targets	19		12	

```
57  else if (error_i == 1'b1)                    0C
58  ncnt = '1;                                   0C
59  else if (abort_i == 1'b1)                    0D
60  ncnt = '1;                                   0D
61  else
62  ncnt = cnt + 1;                              1
63  end
64  aborted: begin                               0
```

Reporting

- 2) Can generate a UCIS compliant XML report file.
- XML file can then be read into another tool such as AsureSign, merged with simulation coverage data to give an overview of whole coverage.
- XML contains pass/fail for all assertions/properties in design and test environment, plus the “structural coverage” information.
- Can thus use formal results to prove bits missed by simulation within the same analysis environment.

Pros and Cons

- Pros
  - Leverages all assertions and properties within environment, whether in RTL or in property environment.
  - Easy understood results which are directly analogous to structural coverage. Doesn't need user to consider all inputs/outputs or restructure properties as you may have to for a completeness check.
  - Easily shows holes in property coverage – easy to identify missing chunks, or chunks missed due to constraints.
  - When merging the coverage in AsureSign, can demonstrate that lines not hit by simulation have been hit formally – coverage is comparable between the two.
- Cons
  - Long property times snowball the quantify run time. Have a property set of around 40 properties, one of which takes 8 hours to run, the others all less than 10 mins. Takes 4 days to run Quantify. Others do run much quicker....
  - As with structural coverage, only shows you've tested what is there, not if any function is missing.

ENERGY EFFICIENCY
COMMUNICATIONS
SECURITY

Innovative semiconductor solutions for energy efficiency, communications and security.