User experiences with open-source mainstream verification techniques

SVA, Unit Testing, RTL Lint, TB linting

Srinivasan Venkataramanan

Deepa Palaniappan

AsFigo

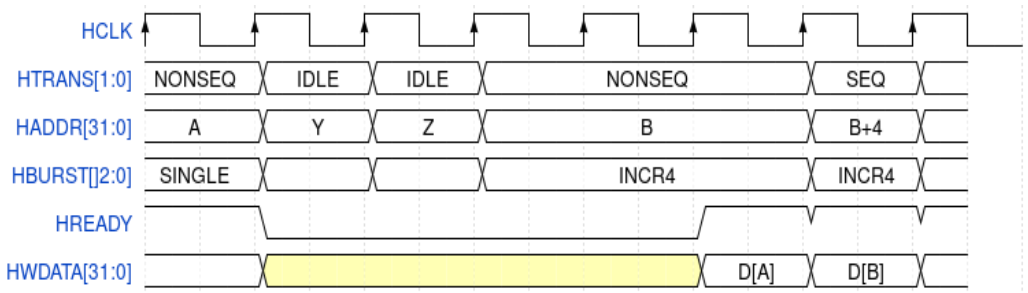
Chip Design as it should be

open source initiative®

Agenda

- Introduction
- Assertion Based Verification - CIP
- Unit Testing
- User experience in porting CIPs to Verilator
- RTL Lint & opensource experience
- Testbench Linting via opensource tools
- Summary, looking forward

AMBA Assertions - plan



. No	Check ID	Requirement
1	a_p_af_psel_pen	penable should appear exactly one clock after psel.
2	a_p_af_idle_paddr_stable	paddr signal should be stable during IDLE transfer.
3	a_p_af_idle_pwrite_stable	pwrite signals should be stable during IDLE transfer.
4	a_p_af_active_one_clk	FSM can stay in ENABLE state for only one clock
5	a_p_af_idle_to_setup	FSM can only move to SETUP state from IDLE state or it can be in IDLE state.
6	a_p_af_paddr_stable_bw_setup_and_en	Ensure paddr to hold between SETUP and ENABLE state.
7	a_p_af_pwdata_stable_bw_setup_and_en	Ensure pwdata to hold between SETUP and ENABLE state.
8	a_p_af_pwrite_stable_bw_setup_and_en	Ensure pwrite to hold between SETUP and ENABLE state.
	a_p_af_pwrite_stable_bw_setup_a	pwdata should not be an unknown value

Chip Design as it should be

AHB LITE MASTER ASSERTION SPECFICATION

S.No	Assertion Specification
1	HTRANS must not be driven to busy after an idle transfer is seen in previous cycle
2	Addresses greater than valid low and high should not be generated
3	Sequential transfers cannot follow IDLE,since the first transfer of any burst must be non_sequential
4	After an IDLE transfer,next clock transfer type be either IDLE or NSEQ
5	When HBURST is of type SINGLE, transfer type must be NSEQ
6	When BUSY cycles are induced within a transfer, Address should not change
	Control signals will be constant within a burst
	Address should not change withn wait states
	Nseq must be followed by seq or burst for incr and wrap bursts
	Controls signals should never be driven to x
	Busy should be followed by a busy or nseq, except for undefined length bursts
	At Reset, htrans should be idle and hresp should be okay
	For Single nseq is never followed by a seq or busy
	For Busy transfer, zero wait state okay response should be seen
	Idle cycles should be followed by a zero wait state okay response

Sample APB assertion

```
a_p_af_active_one_clk : assert property
(disable iff (!present)
af_apb_acc_st |=>
  af_apb_setup_st || af_apb_idle_st))
else
  `AF_CIP_ERROR
  ("a_p_af_active_one_clk"
   "FSM stuck in ENABLE state");
```

Abort cond

Consequent

Anteceden t

Fail action block

SVA support in Verilator

```
property p_vw_low_power ;
   vw_apb_idle_st |-> $stable (paddr);
endproperty : p_vw_low_power
a_p_vw_low_power: assert property (p_vw_low_power )
```

```
(psel && !penable) |=> penable)
```

```
$countones, $isunknown, $onehot0
```

```
default clocking dcb @(posedge pclk);
endclocking : dcb
```

Feature	Comments
Named Properties	Yes (Not when we started, raised a ticket and Verilator fixed it)
Verification directives	Yes
Signal Value functions	Yes
Implication operator	Yes
Default Clocking	Yes

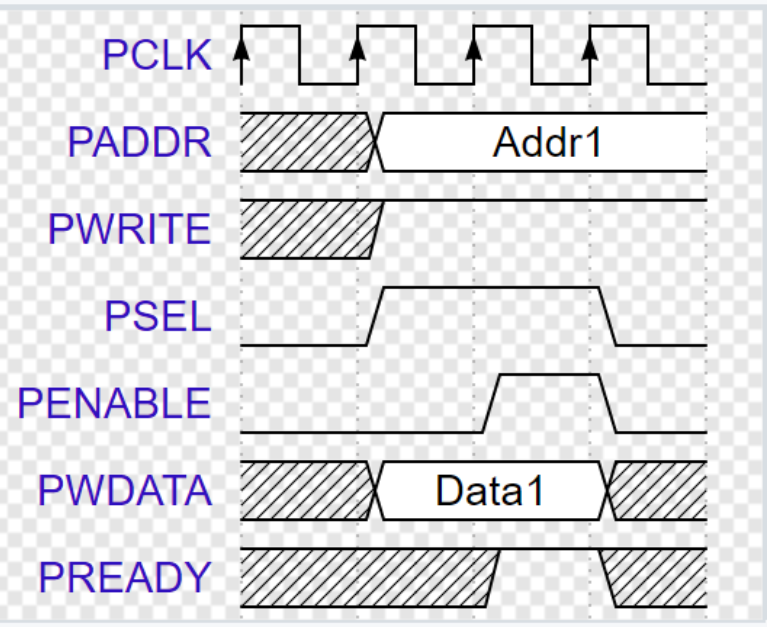
Feature	Comments
Default Disable	No, but should be easy
Sequence	No, complex

Unit Testing

- Easier than full UVM
- Write Unit test per assertion/feature
  - Much easier to “test”
  - PASS and FAIL tests
  - Use SVUnit

SVUnit
a new chance to lead

Unit Tests for APB assertion

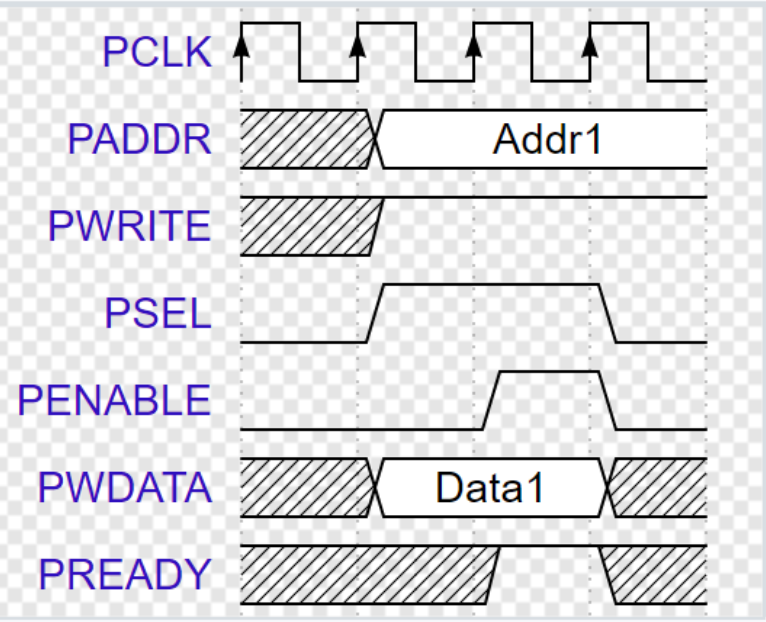



```
psel && !pen
|=> pen
```

- Pass trace:
  - psel && !pen
  - After 1 clock, driven pen = 1
  - SVA should pass
- Fail trace:
  - psel && !pen
  - After 1 clock, driven pen = 0
  - SVA should FAIL

AsFigo
Chip Design as it should be

Unit Tests for APB assertion



psel && !pen |=> pen

```
`SVTEST(fail_a_p_af_psel_pen)
   `g2u_display ("fail_a_p_af_psel_pen")
   idle ();
   `g2u_display ("Driving psel to 1")
   psel = 1'b1;
   wait_for_n_clks (1);
   `g2u_display ("Driving pen to 0")
   penable = 1'b0;
   wait_for_n_clks (1);
   penable = 1'b0;
   idle ();
   wait_for_n_clks (10);
   `FAIL_IF_LOG(1, "Expected FAIL a_p_af_psel_pen")
   `g2u_display ("End fail_a_p_af_psel_pen")
 `SVTEST_END
```

AsFigo
Chip Design as it should be

Complete Testsuite - APB

SVUnit
a new chance to lead

- Developed 2 tests per assertion
- Group Unit-tests under “TESTSUITE”

Num Unit tests	PASS count	FAIL (intended, -ve tests)
26	13	13

AsFigo
Chip Design as it should be

Our contributions – GitHub repos/projects

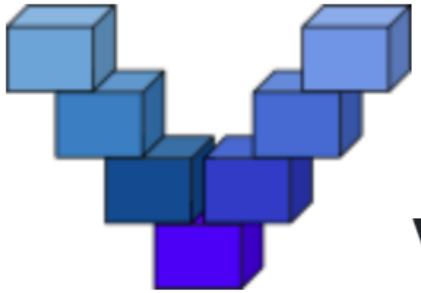
- Used many opensource projects
  - Verilator
  - SVUnit
  - Verible
  - sv_waveterm
- Ported CIP to Verilator
  - Workarounds added
  - Reported 10+ issues
  - Fixed one inside Verilator source
- Ported SVUnit to Verilator
  - 4+ issues, branch PR
- Ported sv_waveterm to Verilator
  - Fixes provided

Project	Issue	Status
SVUnit	SVUnit ported to Verilator	Fixed
sv_wateter	Fixes for Verilator, added Makefile for Verilator & Riviera-PRO #3	Fixed
Verilator	SV: default clocking leads to syntax error later #4032	Fixed
Verilator	SV: $info severity wrongly printed as "Assertion failed" #4036	Fixed
Verilator	Support function call without ()	Fixed
Verilator	SVA: named property not working in latest/devel version? #3879	Fixed

12	Verilator	SVA: Support for default disable iff #4016	Open
13	Verilator	SVA: Delay at start of property definition does not compile #4017	Open
14	Verible	SVA: syntax error around fail-action block under ifdef #1792	Open
15	Verilator	$monitor - ignore $time changes (LRM compliant) #4040	Open
16	Verilator	SVA: Controlling vacuous success #4038	Open
17	Verilator	Interface + generate begin does not compile #4065	Open
18	Verilator	SVA checker..endchecker support #4066	Open

RTL Lint - opensource

- RTL lint is very popular
  - Naming guidelines
  - Coding style checks
  - Synthesis checks
  - DFT checks
- Many teams even integrate this to
- Good opensource alternates becor



Google

Verible

lowRISC RTL guidelines + Opensource

- lowRISC – a popular opensource SoC company
- Provides: openTitan SoC
- Published a set of rules for RTL code
- Our team is working on Linting those rules with opensource tools:
  - Verilator
  - Verible and more

AsFigo

Chip Design as it should be

Implementing lowRISC rules using open-source tools

- Both Verible & Verilator tried
- Easy to integrate with CI/CD
- Custom policies setup
  - Easy in Verible
  - New issue opened with Verilator for use-model
  - `verilator_config
- Potential candidate for primetime adoption!

lowRISC rule	Verible	Verilator
Bit vectors and packed arrays must be little-endian.	packed-dimensions-range-ordering	ASCRANGE
Declare all signals	NONE	IMPLICIT
Functions	explicit-function-lifetime	IMPLICITSTATIC
Name your generated blocks	generate-label	NONE

Testbench Linting

- New paradigm
  - Old/proven concept
  - Lack of good parsers/tools/API
- Verible – C++ based rules
- Svlint – Rust based, custom plugin
- PySlint – Python based, works on top of slang/pyslang
- UVMLint – open-source Linter for UVM code

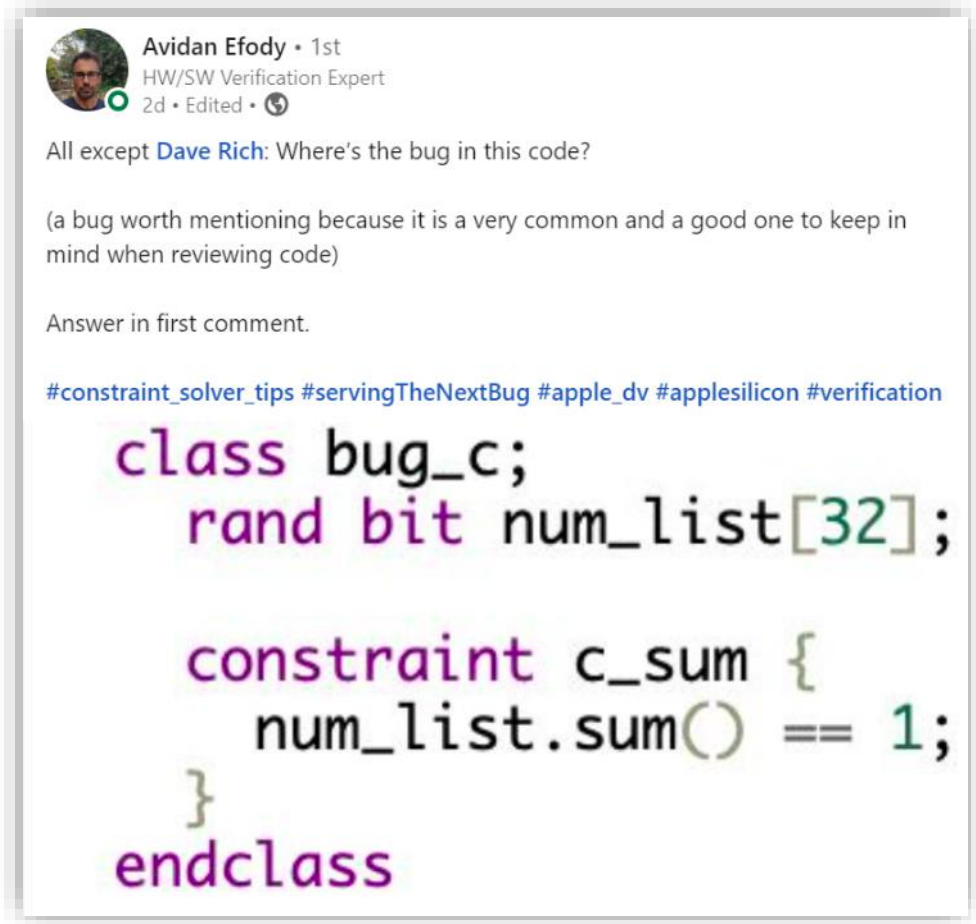
Testbench Linting – Sample rules

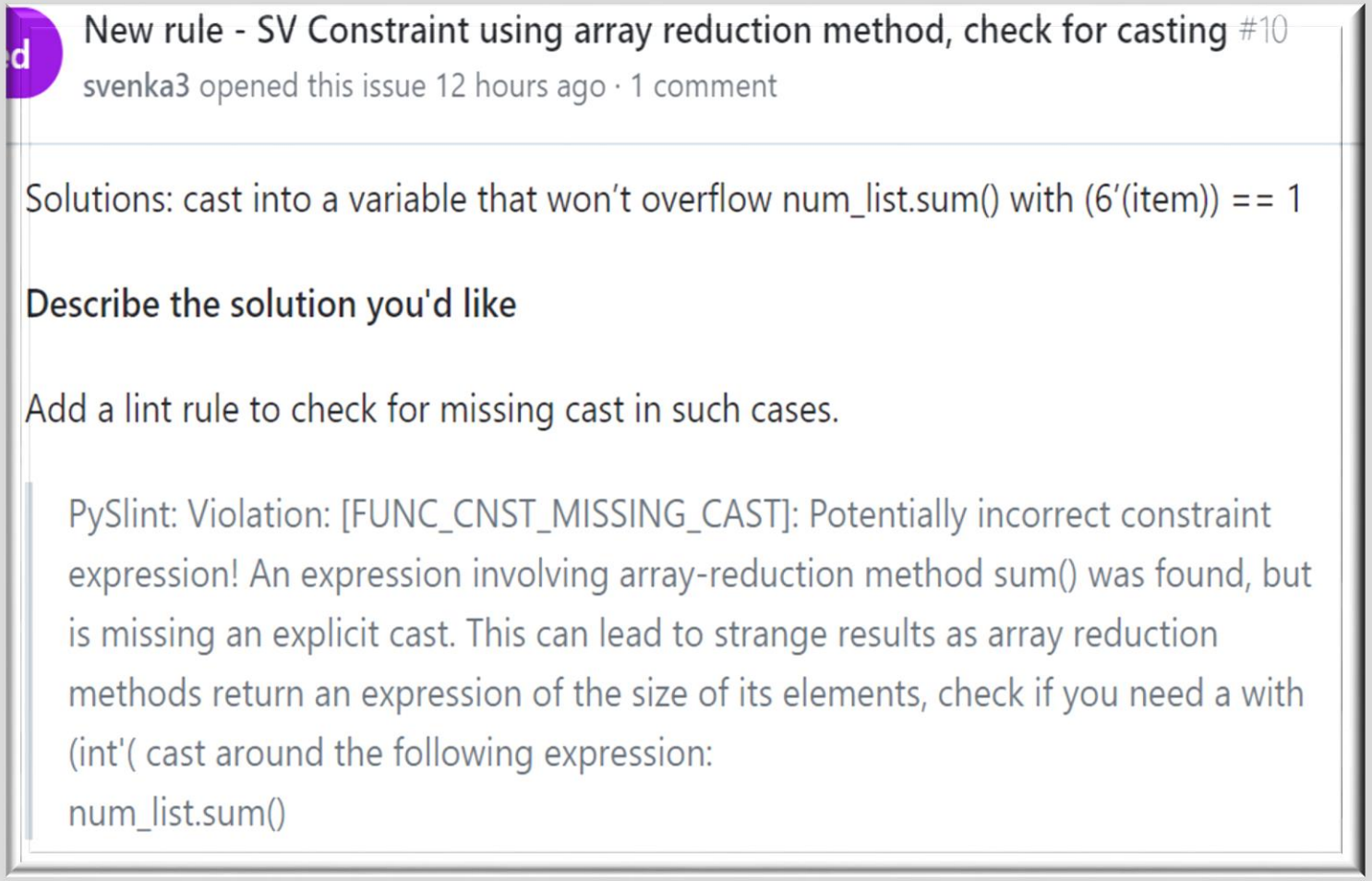
PySlint Rule ID	Description
NAME_CL_PREFIX NAME_CG_PREFIX NAME_ASM_PREFIX NAME_COV_PREFIX	Naming/style checks
SVA_MISSING_LABEL SVA_MISSING_ENDLABEL SVA_NO_PASS_AB SVA_MISSING_FAIL_AB	Assertion specific. Performance, Debug categories
CL_METHOD_NOT_EXTERN CL_MISSING_ENDLABEL FUNC_CNST_MISSING_CAST	Class specific – style, functionality

UVMLint Rule ID	Description
UVM_DRV_MISSING_PARAMS	Driver should have REQ parameter overridden
UVM_AGENT_IS_ACT_HIDE	Do not re-declare is_active field in uvm_agent
UVM_MON_MISSING_VIF	Monitor should have a handle to virtual interface

AsFigo
Chip Design as it should be

SV Constraints – case study





AsFigo
Chip Design as it should be

Testbench Linting - summary

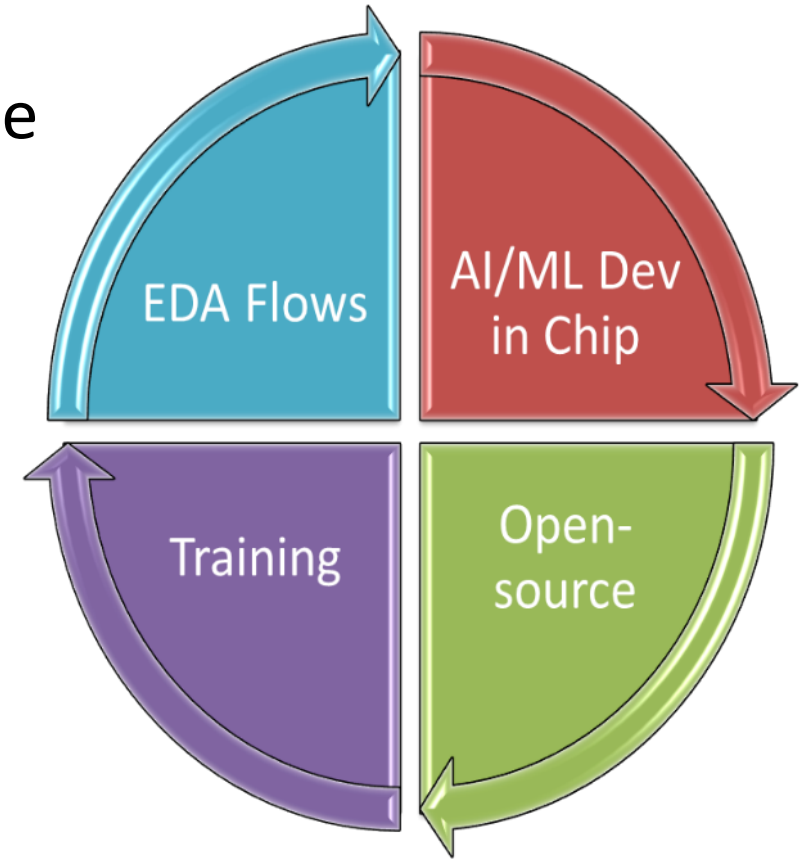
- Exciting new frontiers
- Opensource can now lead the way for commercial tools to follow
  - Than just playing a catch-up game
  - Faster turn-around time
  - Potentially vast set of developers!
- UVMLint
  - WIP linter for UVM code
  - First set of rules now available via GitHub!
  - Looking for volunteers and contributors to enhance

Summary, next steps

- Open-source verification is now real
- Shared our experience in:
  - Assertions
  - Unit Testing
  - RTL Linting
  - Testbench Linting
- PySlint – a case where open-source can leapfrog beyond commercial tools
  - Opportunity for UK Universities to take lead!

Who we are, what do we do?

- UK based chip design start-up
- Bringing decades of Chip Design and EDA expertise
- Globally reputed training solutions
  - RTL Design, Verification
  - SystemVerilog, VHDL, e, PSL etc.
  - Methodologies: UVM, OVM, VMM, eRM etc.
- Bringing AI/ML to chip design flows
  - Generative AI
  - AI Analytics



AsFigo
Chip Design as it should be

Thanks, Q&A

AsFigo

Chip Design as it should be