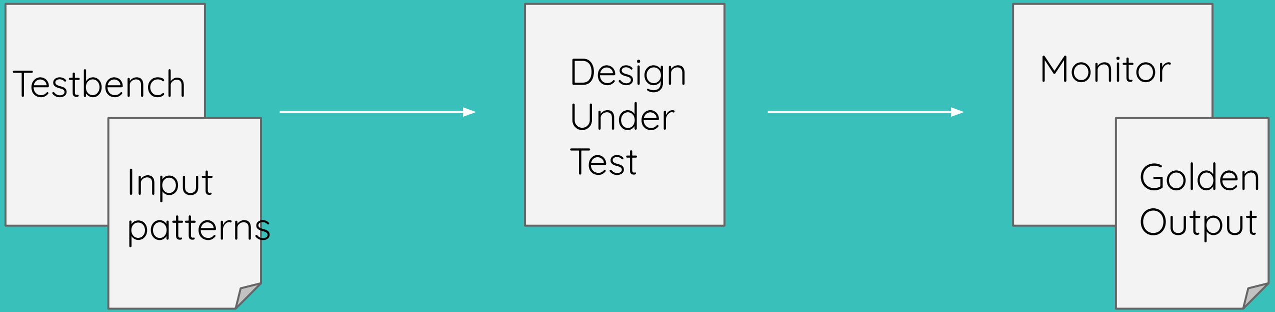
Indian Institute of Technology Gandhinagar

Formal Verification of RISC-V Processor

Presented by:
Shruti Prakash Gupta

VS

Direct Verification



Testbench

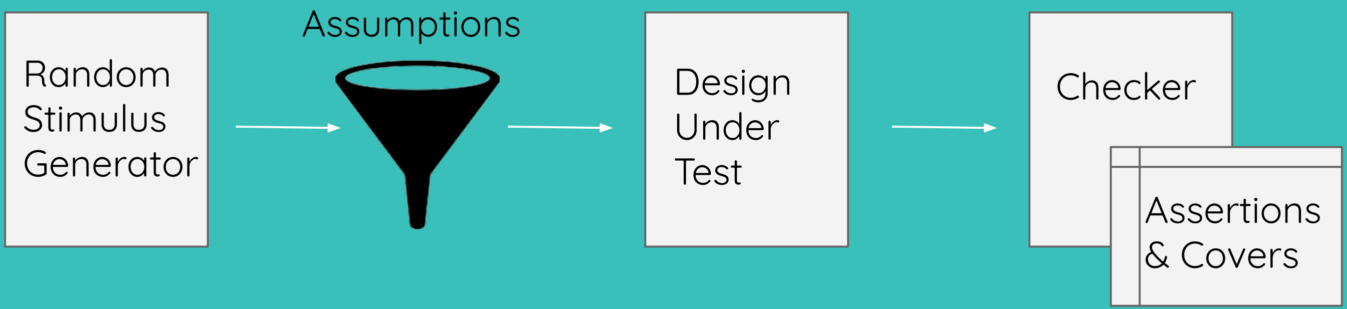
Input patterns

Design Under Test

Monitor

Golden Output

Formal Verification



Random Stimulus Generator

Assumptions

Design Under Test

Checker

Assertions & Covers

Formal Verification : Introduction

Formal Verification Technique

1. A technique to validate the functional correctness of (hardware) designs.
2. Requires mathematical model of the system
3. Exercises all possible inputs and checks validity of outputs generated

FV Tools and Environment

1. Generate constrained-random test stimulus
2. Ensure 100% test coverage for the given inputs
3. Validate the provided properties (assertions) - derived from the design specifications

1

Formal Verification Test Environment

- System Verilog Constructs and Semantics
- Jasper Gold Environment
- Report Format and Analysis

System Verilog Constructs and Semantics

SV Property:

- `property (@(condition) a |=> b);`
- Includes trigger signal (or condition) & behaviour

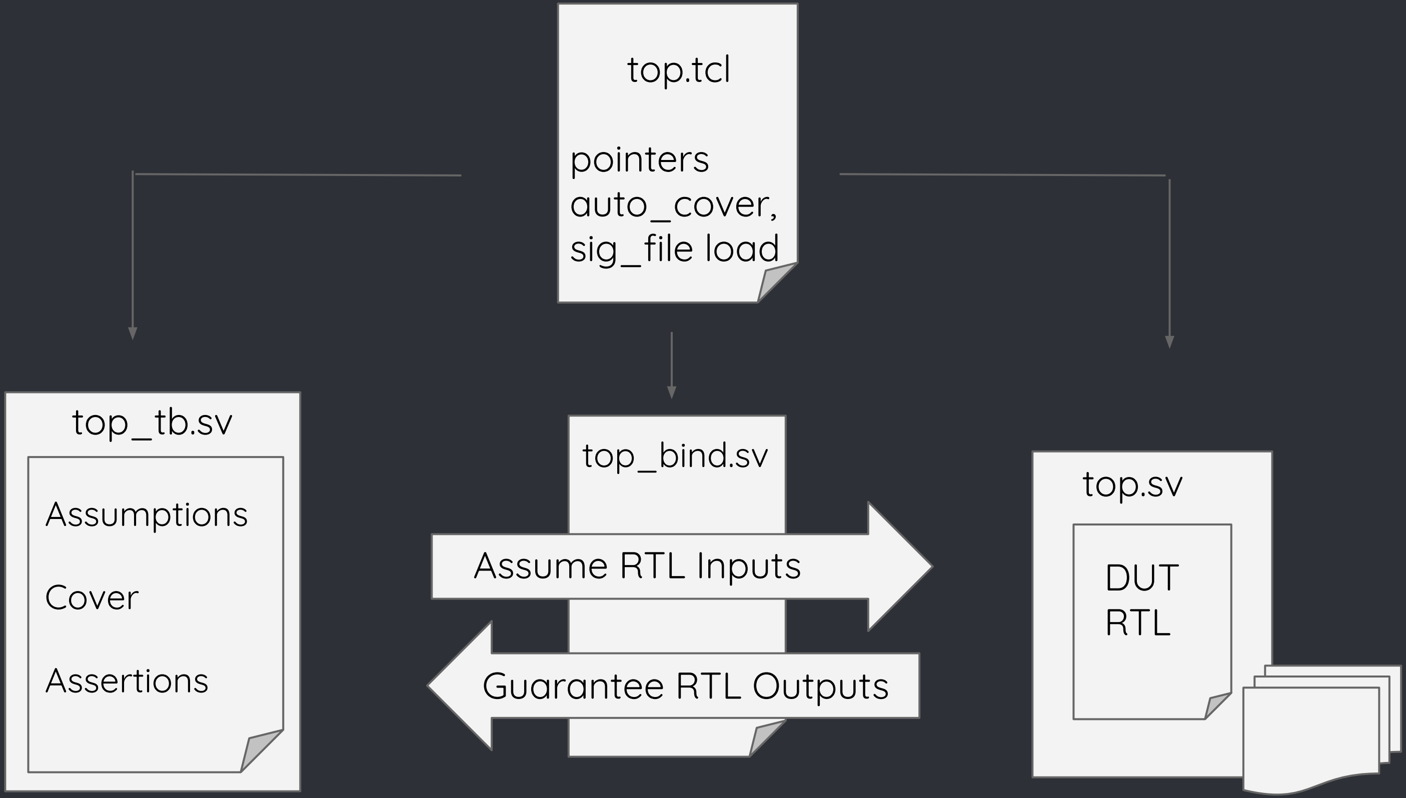
Concurrent Assumptions & Assertions:

- `property (@(posedge clk) disable iff (reset) a ##2 b |=> c);`
- Triggered at clock edge & disabled at reset
- May include delay constructs
- Assumption - Restricts & Assertion - Validates

SV Covers:

- `property (@(posedge clk) disable iff (reset) a ##2 b);`
- Used to witness a condition - sanity check

Jasper Gold Test Environment



top.tcl

pointers auto_cover, sig_file load

top_tb.sv

Assumptions

Cover

Assertions

top_bind.sv

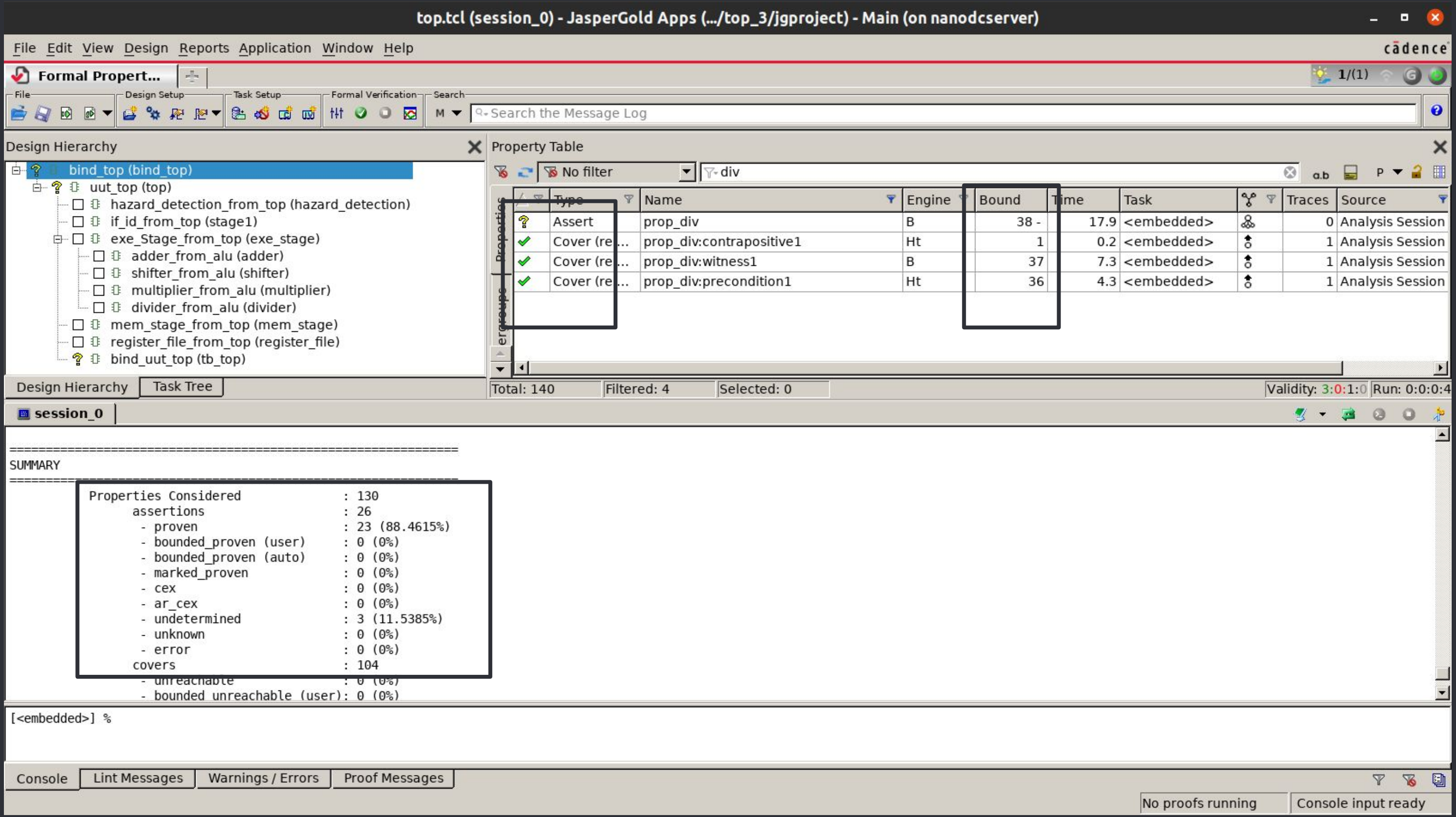
Assume RTL Inputs

Guarantee RTL Outputs

top.sv

DUT RTL

Report Format and Analysis



```
bind_top (bind_top)
  uut_top (top)
    hazard_detection_from_top (hazard_detection)
    if_id_from_top (stage1)
    exe_Stage_from_top (exe_stage)
      adder_from_alu (adder)
      shifter_from_alu (shifter)
      multiplier_from_alu (multiplier)
      divider_from_alu (divider)
    mem_stage_from_top (mem_stage)
    register_file_from_top (register_file)
    bind_uut_top (tb_top)
```

	Type	Name	Engine	Bound	Time	Task		Traces	Source
?	Assert	prop_div	B	38 -	17.9	<embedded>		0	Analysis Session
✓	Cover (re...	prop_div:contrapositive1	Ht	1	0.2	<embedded>		1	Analysis Session
✓	Cover (re...	prop_div:witness1	B	37	7.3	<embedded>		1	Analysis Session
✓	Cover (re...	prop_div:precondition1	Ht	36	4.3	<embedded>		1	Analysis Session

Total: 140 Filtered: 4 Selected: 0 Validity: 3:0:1:0 Run: 0:0:0:4

```
==============================================================
SUMMARY
==============================================================
           Properties Considered              : 130
                 assertions                   : 26
                  - proven                    : 23 (88.4615%)
                  - bounded_proven (user)     : 0 (0%)
                  - bounded_proven (auto)     : 0 (0%)
                  - marked_proven             : 0 (0%)
                  - cex                       : 0 (0%)
                  - ar_cex                    : 0 (0%)
                  - undetermined              : 3 (11.5385%)
                  - unknown                   : 0 (0%)
                  - error                     : 0 (0%)
                 covers                       : 104
                  - unreachable               : 0 (0%)
                  - bounded_unreachable (user): 0 (0%)
```

```
[<embedded>] %
```

No proofs running Console input ready

2

Design Under Test & Test Methodology

- RISC V - RV32IM
- Pipelined Structure of DUT
- Stages with Abstraction for Verification
- Combinational Block Verification with FV
- Assume - Guarantee Method

RISC V - RV32IM

- ➢ This Processor is based on RISC V 32 bit ISA
- ➢ Both Data and Instruction sizes are 32-bit
- ➢ IM extension – Standard Extension for Integer Multiplication and Division
- ➢ Following Instructions Supported

Register Type

- ADD, SUB
- SLT, SLTU
- XOR, AND, OR
- SLL, SRL, SRA
- MUL, MULH, MULHSU, MULHU
- DIV, DIVU, REM, REMU

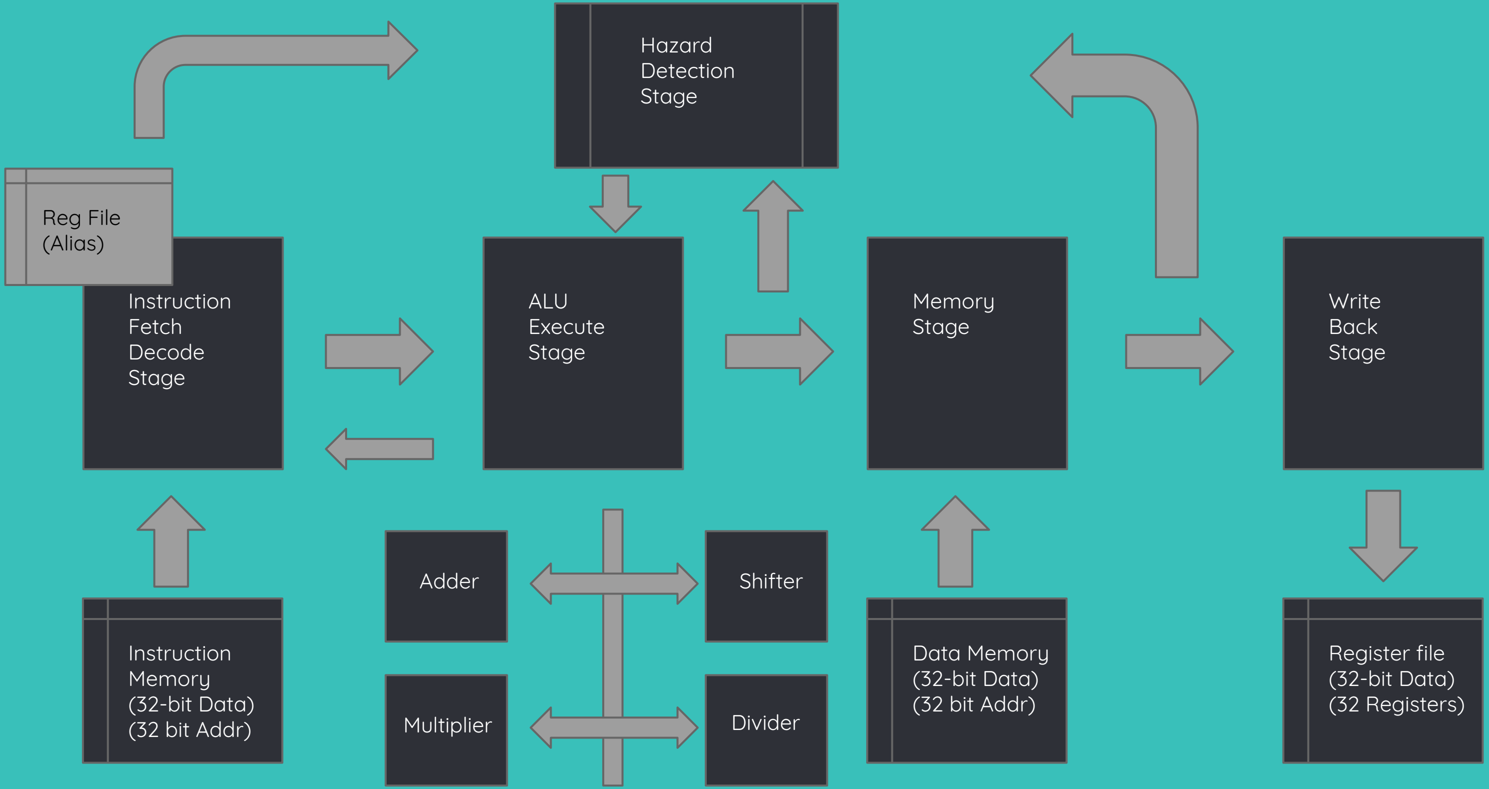
Immediate Type

- ADDI,
- SLTI, SLTIU
- XORI, ORI, ANDI
- SLLI, SRLI, SRAI

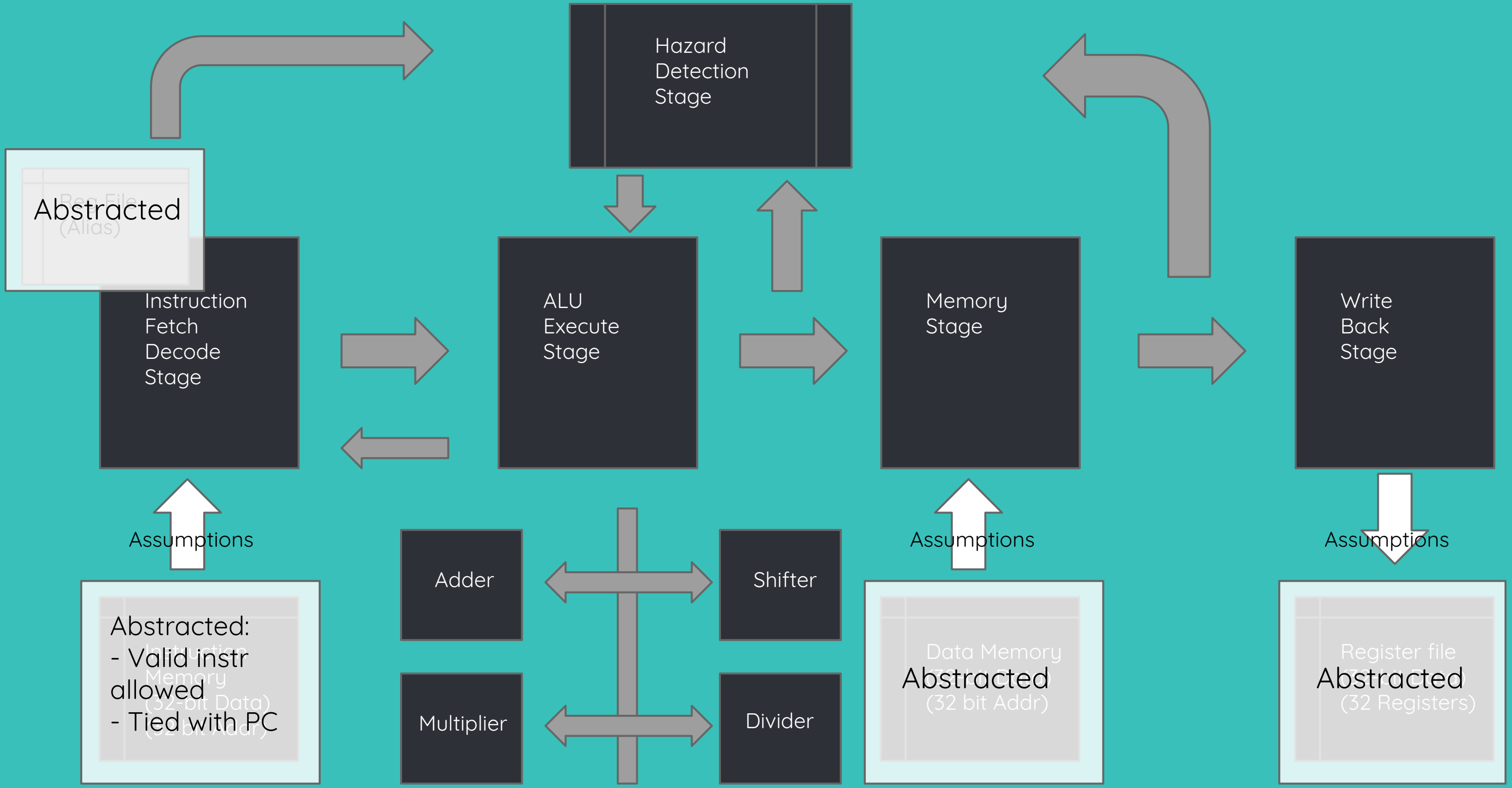
Load, Store & Branch

- LB, LH, LW
- LBU, LHU
- SB, SH, SW
- BEQ, BNE
- BLT, BGE
- BLTU, BGEU
- LUI, AUIPC
- JAL, JALR

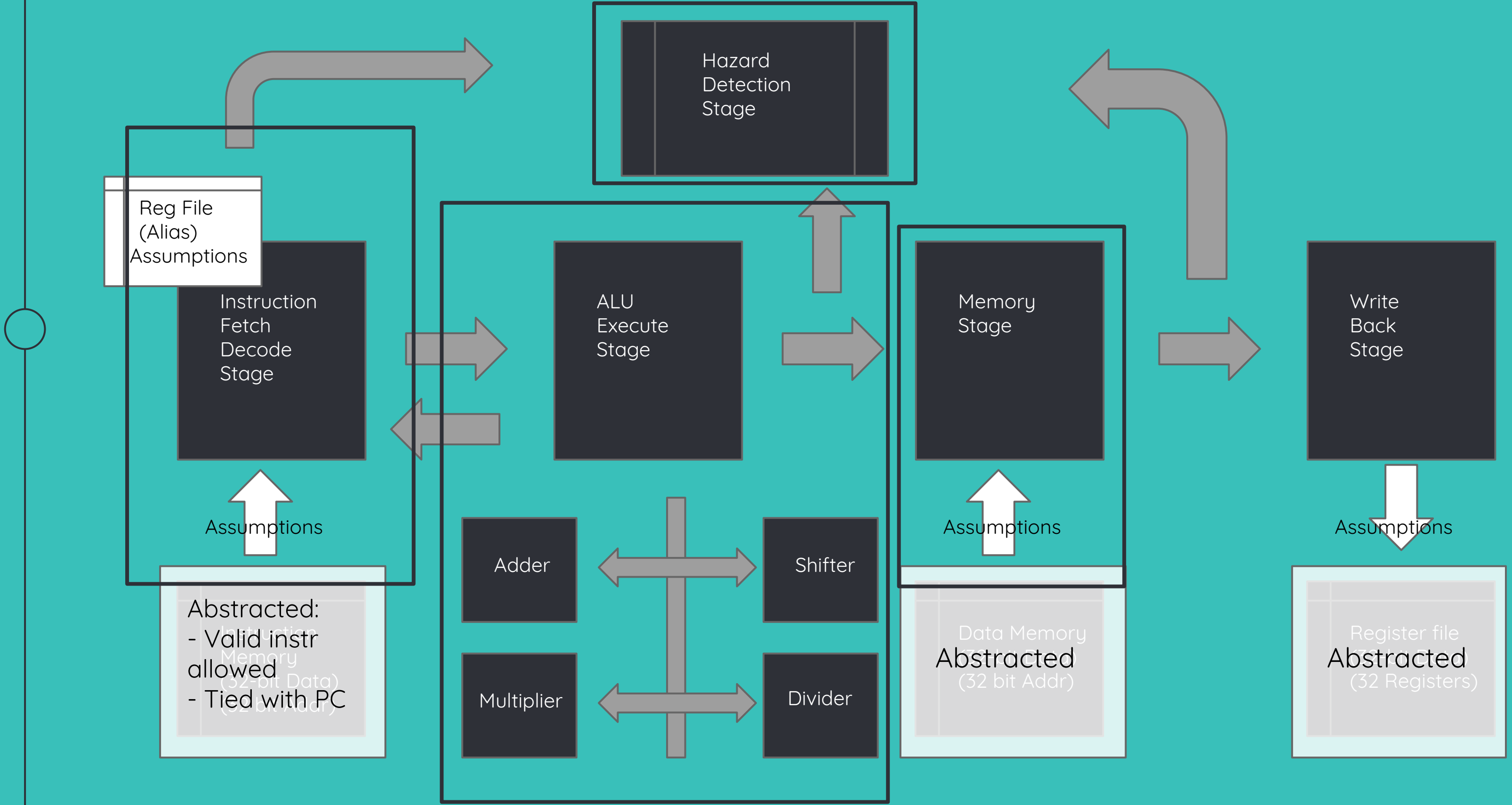
DUT Pipeline


Hazard Detection Stage
Reg File (Alias)
Instruction Fetch Decode Stage
ALU Execute Stage
Memory Stage
Write Back Stage
Adder
Shifter
Multiplier
Divider
Instruction Memory (32-bit Data) (32 bit Addr)
Data Memory (32-bit Data) (32 bit Addr)
Register file (32-bit Data) (32 Registers)

Verification Method : Abstraction



Assume Guarantee



Hazard Detection Stage

Reg File (Alias) Assumptions

Instruction Fetch Decode Stage

ALU Execute Stage

Memory Stage

Write Back Stage

Assumptions

Assumptions

Assumptions

Adder

Shifter

Multiplier

Divider

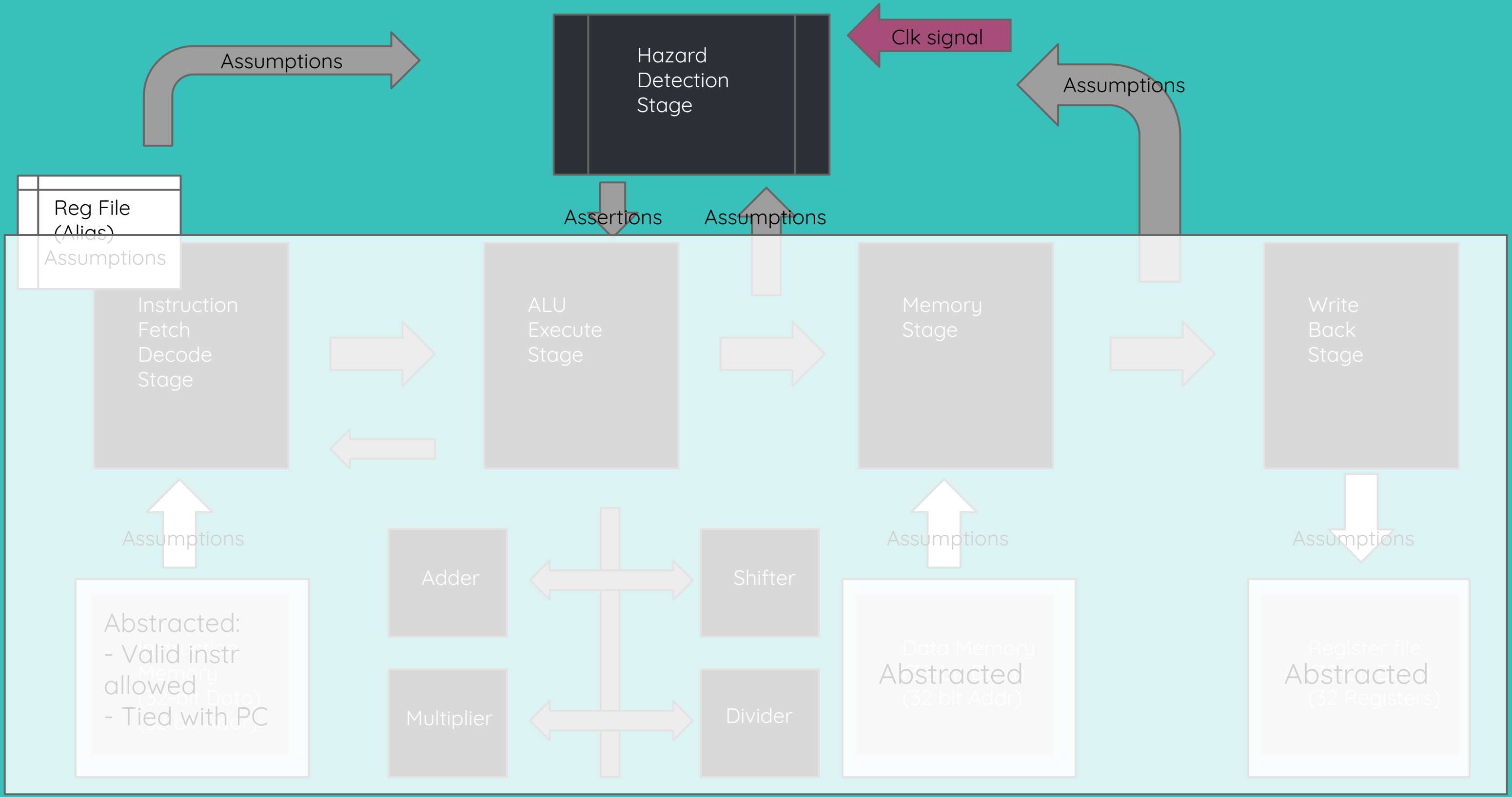
Abstracted:
- Valid instr allowed
- Tied with PC

Instruction Memory (32-bit Data) (32 bit Addr)

Data Memory Abstracted (32 bit Addr)

Register file Abstracted (32 Registers)

Combinational Block Verif with FV



3

Debugging and Verification with FV

- Assumption and Assertion Highlights
- Report Analysis - Proven & Undetermined Properties
- Inconclusive Property Depth Analysis
- Debugging & Design with FV

Assumptions & Assertions Highlights

1. pc_inc_prop:

```
(!(branch_taken_w || (id_alu_op_r == `ALU_DIV)), pc_curr = iaddr_o)
|=> iaddr_o == (pc_curr + 4);
```

2. prop_ex_hazard_b:

```
(( (id_rb_index_w == ex_rd_index_r)  &&  (id_rb_index_w != 5′d0) ),
rb_val_1 = ex_alu_res_r) |=> (exe_rb_r == rb_val_1);
```

3. prop_br_br_not_allowed:

```
@(posedge clk_i) disable iff (reset_i) branch_taken_w |=> !branch_taken_w;
```

Assumptions & Assertions Highlights

4. prop_beq_fail:

```
(id_branch_r == `BR_EQ) && ( ! taken(id_ra_value_r, id_rb_value_r, id_imm_r,
id_op_imm_r, `BR_EQ, branch_taken_w))
|=> !branch_taken_w;
```

5. prop_or:

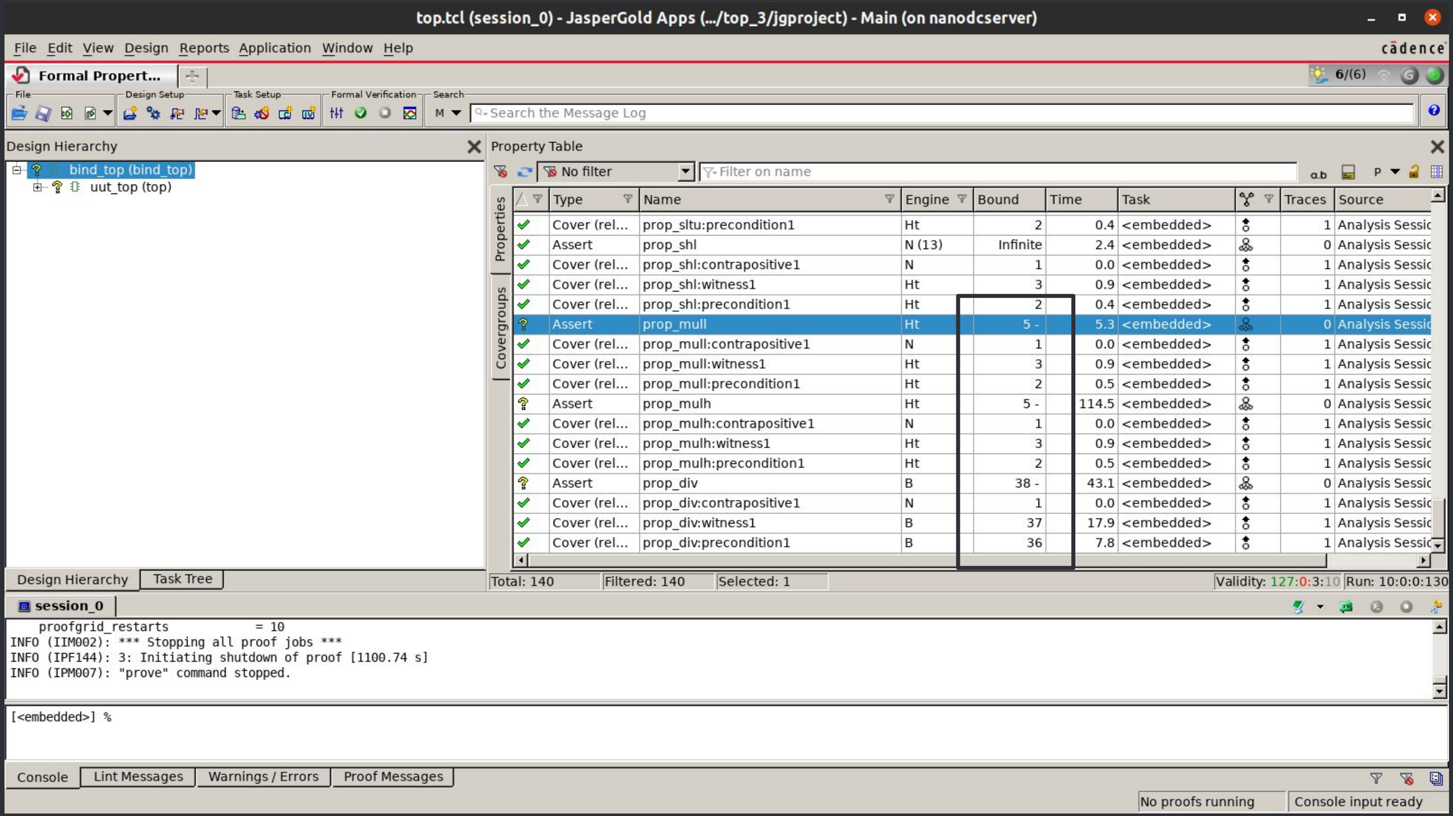
```
(( (id_alu_op_r == `ALU_OR)  &&  (!branch_taken_w) ),
result_or = alu_out (id_ra_value_r, id_rb_value_r, id_a_signed_r,
id_b_signed_r, id_imm_r, id_op_imm_r, id_alu_op_r, id_next_pc_r))
|=> (ex_alu_res_r == result_or);
```

Proven & Undetermined Properties

Individual Blocks Active

Stage	# of Assumptions	# of Assertions	# of Covers	# of Undetermined Properties	Max time for Convergence (in sec)	Failures Encountered while Verif
IF/ID Stage	**13**	**6**	**52**	**0**	**0.4**	**1. Stall signal design** **2. Wire vs Reg Error**
Execute Stage	**27**	**26**	**153**	**3**	**7.5**	**1. Divide operation errors - special cases & stall** **2. Stall signal does not affect PC**
Memory Stage	**1**	**4**	**13**	**0**	**0.0**	**1. Incorrect value passed to hazard detect stage**
Hazard Detection Stage	**5**	**6**	**24**	**0**	**0.2**	**1. Memory word error - selection between memory & execute stage value**

Inconclusive Property Depth Analysis



	Type	Name	Engine	Bound	Time	Task	Traces	Source
✓	Cover (rel...	prop_sltu:precondition1	Ht	2	0.4	<embedded>	1	Analysis Sessic
✓	Assert	prop_shl	N (13)	Infinite	2.4	<embedded>	0	Analysis Sessic
✓	Cover (rel...	prop_shl:contrapositive1	N	1	0.0	<embedded>	1	Analysis Sessic
✓	Cover (rel...	prop_shl:witness1	Ht	3	0.9	<embedded>	1	Analysis Sessic
✓	Cover (rel...	prop_shl:precondition1	Ht	2	0.4	<embedded>	1	Analysis Sessic
?	Assert	prop_mull	Ht	5 -	5.3	<embedded>	0	Analysis Sessic
✓	Cover (rel...	prop_mull:contrapositive1	N	1	0.0	<embedded>	1	Analysis Sessic
✓	Cover (rel...	prop_mull:witness1	Ht	3	0.9	<embedded>	1	Analysis Sessic
✓	Cover (rel...	prop_mull:precondition1	Ht	2	0.5	<embedded>	1	Analysis Sessic
?	Assert	prop_mulh	Ht	5 -	114.5	<embedded>	0	Analysis Sessic
✓	Cover (rel...	prop_mulh:contrapositive1	N	1	0.0	<embedded>	1	Analysis Sessic
✓	Cover (rel...	prop_mulh:witness1	Ht	3	0.9	<embedded>	1	Analysis Sessic
✓	Cover (rel...	prop_mulh:precondition1	Ht	2	0.5	<embedded>	1	Analysis Sessic
?	Assert	prop_div	B	38 -	43.1	<embedded>	0	Analysis Sessic
✓	Cover (rel...	prop_div:contrapositive1	N	1	0.0	<embedded>	1	Analysis Sessic
✓	Cover (rel...	prop_div:witness1	B	37	17.9	<embedded>	1	Analysis Sessic
✓	Cover (rel...	prop_div:precondition1	B	36	7.8	<embedded>	1	Analysis Sessic

Total: 140 Filtered: 140 Selected: 1 Validity: 127:0:3:10 Run: 10:0:0:130

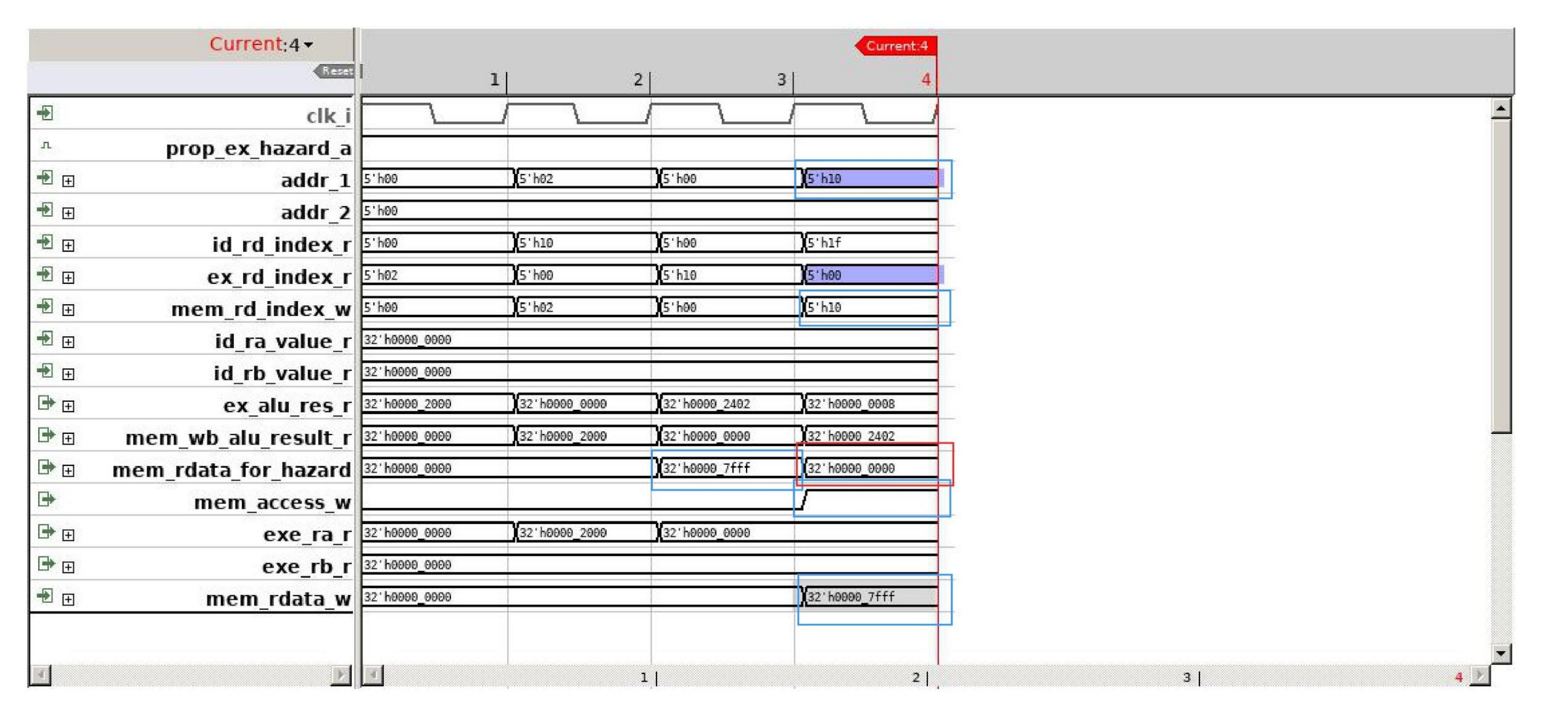
```
    proofgrid_restarts             = 10
INFO (IIM002): *** Stopping all proof jobs ***
INFO (IPF144): 3: Initiating shutdown of proof [1100.74 s]
INFO (IPM007): "prove" command stopped.
```

```
[<embedded>] %
```

No proofs running Console input ready

Debugging & Design with FV

Wire vs Register - Value reaching at different instances


Current:4
Reset
1
2
3
4
Current:4
clk_i
prop_ex_hazard_a
addr_1 5'h00 5'h02 5'h00 5'h10
addr_2 5'h00
id_rd_index_r 5'h00 5'h10 5'h00 5'h1f
ex_rd_index_r 5'h02 5'h00 5'h10 5'h00
mem_rd_index_w 5'h00 5'h02 5'h00 5'h10
id_ra_value_r 32'h0000_0000
id_rb_value_r 32'h0000_0000
ex_alu_res_r 32'h0000_2000 32'h0000_0000 32'h0000_2402 32'h0000_0008
mem_wb_alu_result_r 32'h0000_0000 32'h0000_2000 32'h0000_0000 32'h0000_2402
mem_rdata_for_hazard 32'h0000_0000 32'h0000_7fff 32'h0000_0000
mem_access_w
exe_ra_r 32'h0000_0000 32'h0000_2000 32'h0000_0000
exe_rb_r 32'h0000_0000
mem_rdata_w 32'h0000_0000 32'h0000_7fff
1
2
3
4

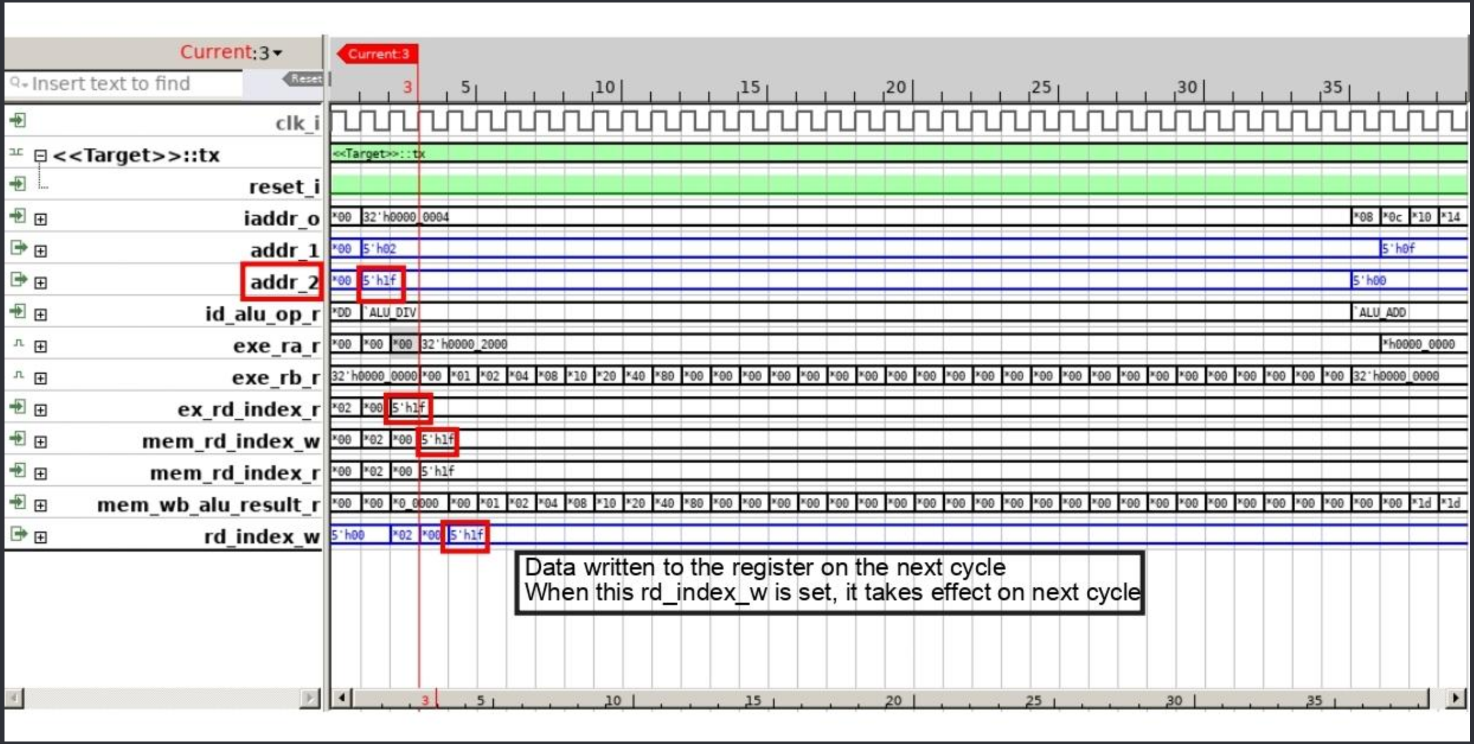
Proven & Undetermined Properties

Complete RTL Active

Stage	# of Assumptions	# of Assertions	# of Covers	# of Undetermined Properties	Max time for Convergence (in sec)	Failures Encountered while Verif
IF/ID Stage	**10**	**6**	**43**	**0**	**6.6**	**1. Stall signal design 2. Extra delay due to WB stage**
Execute Stage	**10**	**26**	**104**	**3**	**114.5**	**1. Divide operation errors - on the fly value change**
Memory Stage	**10**	**4**	**38**	**0**	**0.8**	**-**
Hazard Detection Stage	**10**	**6**	**44**	**0**	**0.2**	**1. Incorrect data passed from curr-2 instr**

Debugging & Design with FV

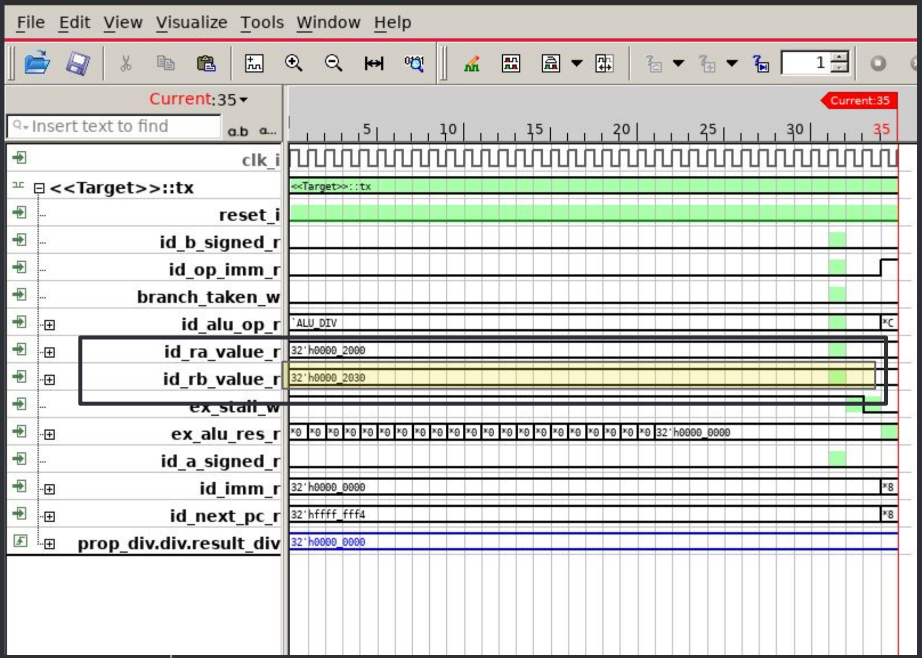
Signal Delayed by one Cycle - Due to extra stage

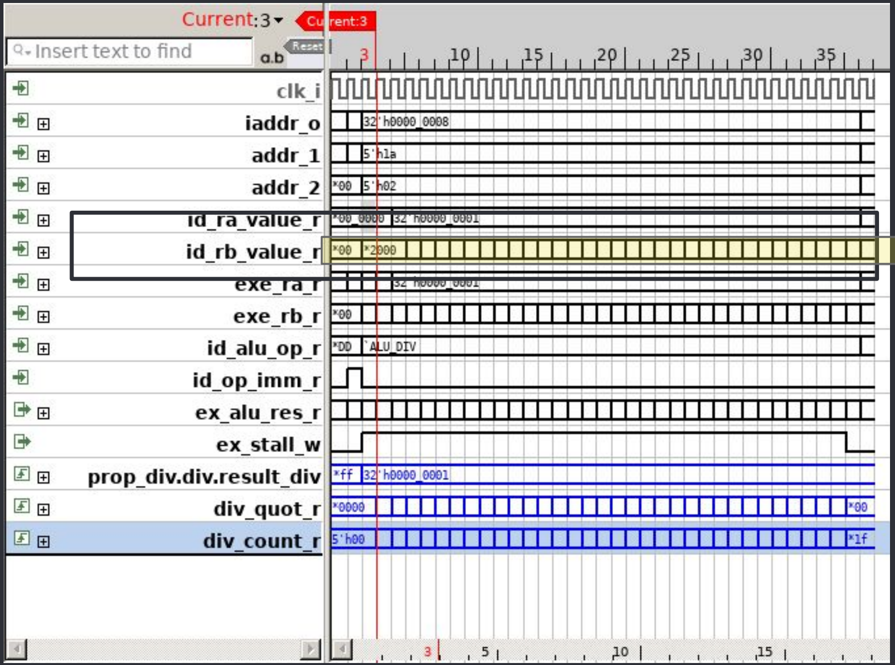


Data written to the register on the next cycle
When this rd_index_w is set, it takes effect on next cycle

Debugging & Design with FV

Unconstrained Signal Changing Value on the Fly



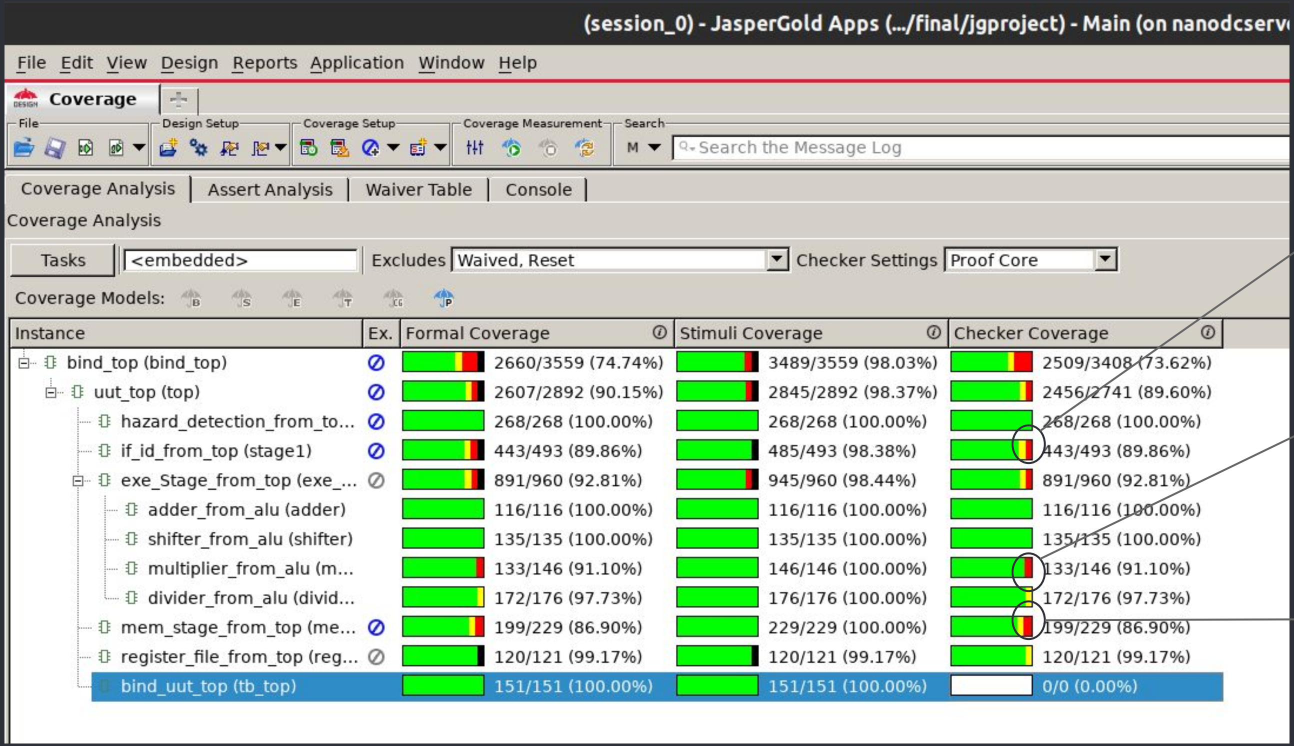


4

Coverage Analysis

- Checking the Completeness of Testbench
  - Jasper Gold Coverage App
  - Coverage Waivers
- Comparison with Direct Verification

Coverage Report with Jasper Coverage App



Tasks <embedded>

Excludes Waived, Reset

Checker Settings Proof Core

Instance	Ex.	Formal Coverage	Stimuli Coverage	Checker Coverage
bind_top (bind_top)	⊘	2660/3559 (74.74%)	3489/3559 (98.03%)	2509/3408 (73.62%)
uut_top (top)	⊘	2607/2892 (90.15%)	2845/2892 (98.37%)	2456/2741 (89.60%)
hazard_detection_from_to...	⊘	268/268 (100.00%)	268/268 (100.00%)	268/268 (100.00%)
if_id_from_top (stage1)	⊘	443/493 (89.86%)	485/493 (98.38%)	443/493 (89.86%)
exe_Stage_from_top (exe_...	⊘	891/960 (92.81%)	945/960 (98.44%)	891/960 (92.81%)
adder_from_alu (adder)		116/116 (100.00%)	116/116 (100.00%)	116/116 (100.00%)
shifter_from_alu (shifter)		135/135 (100.00%)	135/135 (100.00%)	135/135 (100.00%)
multiplier_from_alu (m...		133/146 (91.10%)	146/146 (100.00%)	133/146 (91.10%)
divider_from_alu (divid...		172/176 (97.73%)	176/176 (100.00%)	172/176 (97.73%)
mem_stage_from_top (me...	⊘	199/229 (86.90%)	229/229 (100.00%)	199/229 (86.90%)
register_file_from_top (reg...	⊘	120/121 (99.17%)	120/121 (99.17%)	120/121 (99.17%)
bind_uut_top (tb_top)		151/151 (100.00%)	151/151 (100.00%)	0/0 (0.00%)

PC bits - deadcode

Multiplier inputs non-exhaustive

Memory data non-exhaustive

Coverage Waivers

Waiver Tags

- Unused Signal (Direct Assignment to Unused Signal)
  - Id_rd_index_r not used in hazard detection stage
- Unused Signal (Data Memory Abstracted)
  - Outputs to the Data memory are not restricted as Data memory is abstracted
- ISA Restrictions
  - Invalid Combinations of Instructions
- Architecture Requirements
  - Deadcode in PC : 2 LSB bits

Comparing Formal with Direct Verification

Direct Verification

- Testbench with direct test cases required
- Simulator (or Reference design) needed to generate Golden Output
- Difficult to debug internal signals
- Undetermined Coverage metric generation

Formal Verification

- 46 Assumptions & 42 Assertions constitute the testbench
- Assertions used to check output validity
- Jasper Gold Environment provides easy signal load-store tracking
- Coverage metric generated with waivers

THANK YOU

References

- RISC-V Instruction Set Manual
- jaspergold_apps_userguide.pdf
- jaspergold_cov_userguide.pdf
- SystemVerilog Assertions (SVA) Constructs
- S. Roy, H. Iwashita & T. Nakata "Formal Verification based on Assume and Guarantee Approach - A Case Study" in Proceedings of ASP-DAC 2000, Asia and South Pacific Design Automation Conference 2000, Yokohama, Japan