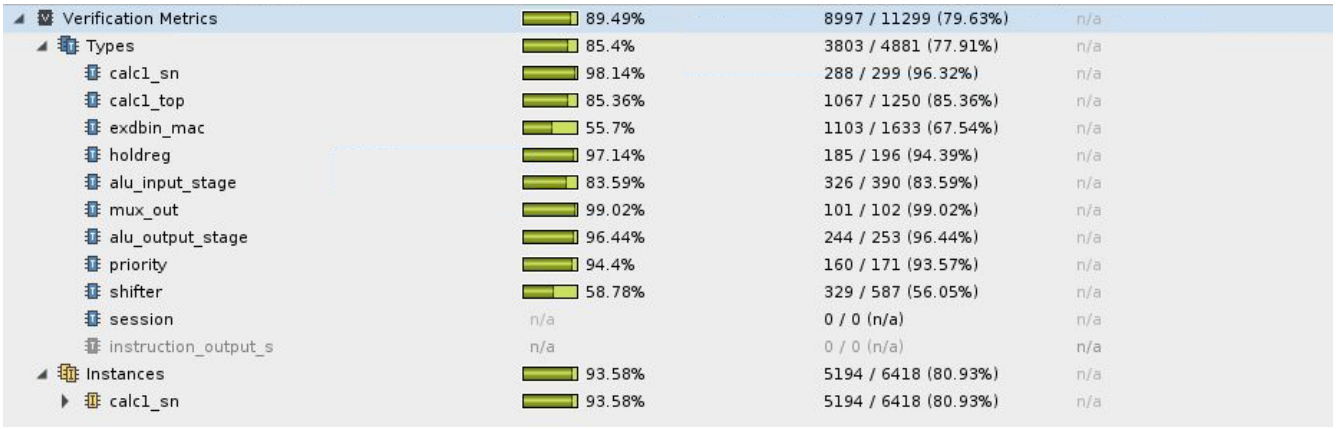
Coverage Report

Jacob Halsey

Code Coverage

The testbench has 89% coverage of the calc1 code:



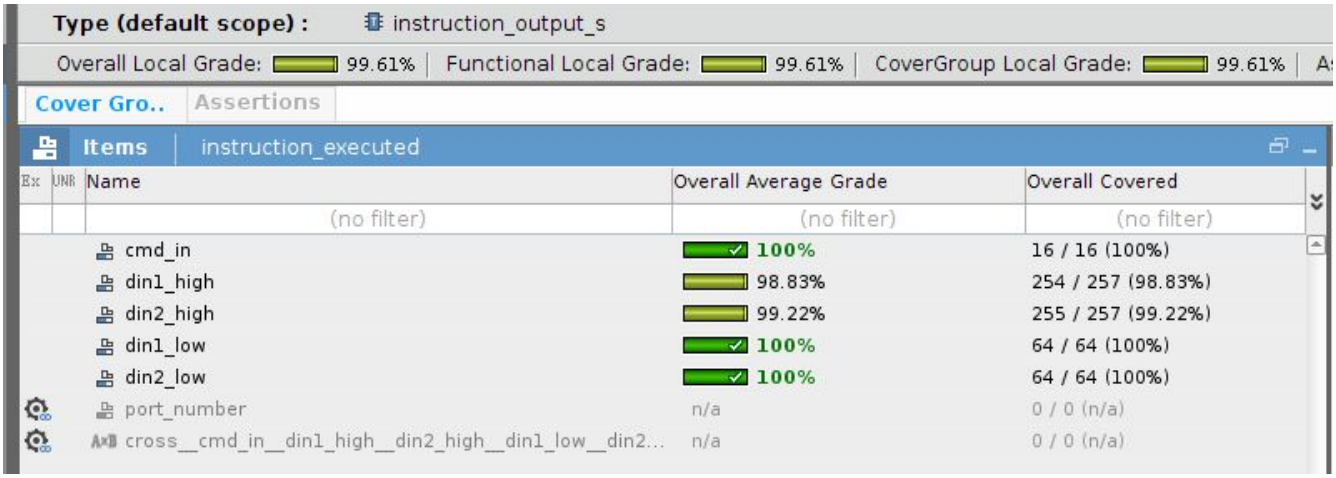
Verification Metrics	89.49%	8997 / 11299 (79.63%)	n/a
Types	85.4%	3803 / 4881 (77.91%)	n/a
calc1_sn	98.14%	288 / 299 (96.32%)	n/a
calc1_top	85.36%	1067 / 1250 (85.36%)	n/a
exdbin_mac	55.7%	1103 / 1633 (67.54%)	n/a
holdreg	97.14%	185 / 196 (94.39%)	n/a
alu_input_stage	83.59%	326 / 390 (83.59%)	n/a
mux_out	99.02%	101 / 102 (99.02%)	n/a
alu_output_stage	96.44%	244 / 253 (96.44%)	n/a
priority	94.4%	160 / 171 (93.57%)	n/a
shifter	58.78%	329 / 587 (56.05%)	n/a
session	n/a	0 / 0 (n/a)	n/a
instruction_output_s	n/a	0 / 0 (n/a)	n/a
Instances	93.58%	5194 / 6418 (80.93%)	n/a
calc1_sn	93.58%	5194 / 6418 (80.93%)	n/a

Functional Coverage

The functional coverage is defined in `coverage.e` as follows:

The cross of the `cmd_in` (opcode), operand1 and operand2 (the high and low bytes of each), and the port number that was used to execute the instruction. By using ranges and buckets for the first and last bytes of the operands it ensures that a large range of numbers are covered, but also that there is variation of the less significant bits as well.

Results show 99% coverage:



Type (default scope) : instruction_output_s

Overall Local Grade: 99.61% | Functional Local Grade: 99.61% | CoverGroup Local Grade: 99.61%

Cover Gro.. | Assertions

Items | instruction_executed

Ex	UNR	Name	Overall Average Grade	Overall Covered
		(no filter)	(no filter)	(no filter)
		cmd_in	100%	16 / 16 (100%)
		din1_high	98.83%	254 / 257 (98.83%)
		din2_high	99.22%	255 / 257 (99.22%)
		din1_low	100%	64 / 64 (100%)
		din2_low	100%	64 / 64 (100%)
		port_number	n/a	0 / 0 (n/a)
		cross__cmd_in__din1_high__din2_high__din1_low__din2...	n/a	0 / 0 (n/a)