Assignment 6

Digital Design Verification

Contents

1 Q1: ALSU

1.1 1. Drive Link for Complete Environment

Here is a link to Complete UVM Environment For ALSU Design.

1.2 2. Verification Plan

Label	Description	Stimulus Generation	Functional Coverage	Functionality Check
ALSU_1	When the reset is asserted, the outputs should be low.	• Directed reset is applied at simulation start via the `do_reset()` task. • Afterwards, the reset signal is randomized with the constraint `c_reset_low_prob` (5% chance of `rst`=1).	• A covergroup in `alsu_rand_class` monitors transitions on `rst`. • Coverage bins ensure that enough reset assertions are observed during randomization.	• The golden model is invoked during reset to verify that both `out` and `leds` are 0. • Any deviation (non-zero outputs when `rst` is high) flags an error.
ALSU_2	In the absence of invalid conditions, when the opcode is ADD, the output should perform addition on ports A and B, incorporating `cin` if `FULL_ADDER` is enabled.	• The random stimulus is generated with the weighted constraint `c_opcode_distribution` to ensure frequent selection of ADD (3'h2). • Inputs A and B are randomized using `c_adder_mult_corner` to emphasize corner cases (values `MAXNEG`, `ZERO`, `MAXPOS`). • Reduction control signals (`red_op_A` and `red_op_B`) are mostly deasserted to avoid invalid conditions.	• The covergroup within `alsu_rand_class` collects data on opcode, A, B, and `cin` along with other control signals. • Specific bins track the occurrence of corner-case operand values and the frequency of the ADD opcode.	• After each randomized transaction, the `golden_model()` task computes the expected output (i.e. `A+B` plus `cin` when relevant). • The testbench compares the DUT output against the golden model; any mismatch in the computed sum flags an error.

Table 1: Verification Plan

1.3 3. Do File

```
vlib work
vlog -f src_files.list +cover -covercells
vsim -voptargs=+acc work.top -classdebug -uvmcontrol=all -cover
do wave.do
coverage save ALSU.ucdb -onexit
run -all
do coverage.do

# to run do file
#-- do run.do
#to execute coverage report
#-- vcover report ALSU.ucdb -details -annotate -all -output code_coverage_rpt.txt -du=ALSU
#-- vcover report -details -cvg -output functional_coverage_report.txt ALSU.ucdb
```

1.4 4. functional Coverage Report

```
Coverage Report by instance with details

=================================================================================
=== Instance: /alsu_coverage_pkg
=== Design Unit: work.alsu_coverage_pkg
=================================================================================

Covergroup Coverage:
    Covergroups                 1        na        na    98.38%
        Coverpoints/Crosses    22        na        na        na
            Covergroup Bins    98        91         7    92.85%

---------------------------------------------------------------------------------
Covergroup                                          Metric       Goal       Bins    Status
---------------------------------------------------------------------------------

TYPE /alsu_coverage_pkg/alsu_coverage/cg            98.38%        100          -    Uncovered
    covered/total bins:                                 91         98          -
    missing/total bins:                                  7         98          -
    % Hit:                                          92.85%        100          -
    Coverpoint rst_cp                              100.00%        100          -    Covered
        covered/total bins:                             2          2          -
        missing/total bins:                             0          2          -
        % Hit:                                     100.00%        100          -
    Coverpoint cin_cp                              100.00%        100          -    Covered
        covered/total bins:                             2          2          -
        missing/total bins:                             0          2          -
        % Hit:                                     100.00%        100          -
    Coverpoint red_op_A_cp                         100.00%        100          -    Covered
        covered/total bins:                             2          2          -
        missing/total bins:                             0          2          -
        % Hit:                                     100.00%        100          -
    Coverpoint red_op_B_cp                         100.00%        100          -    Covered
        covered/total bins:                             2          2          -
        missing/total bins:                             0          2          -
        % Hit:                                     100.00%        100          -
```

Coverpoint bypass_A_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
Coverpoint bypass_B_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
Coverpoint direction_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
Coverpoint serial_in_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
Coverpoint A_cp	100.00%	100	–	Covered
covered/total bins:	5	5	–	
missing/total bins:	0	5	–	
% Hit:	100.00%	100	–	
Coverpoint B_cp	100.00%	100	–	Covered
covered/total bins:	5	5	–	
missing/total bins:	0	5	–	
% Hit:	100.00%	100	–	
Coverpoint A_cp_mod	100.00%	100	–	Covered
covered/total bins:	3	3	–	
missing/total bins:	0	3	–	
% Hit:	100.00%	100	–	
Coverpoint B_cp_mod	100.00%	100	–	Covered
covered/total bins:	3	3	–	
missing/total bins:	0	3	–	
% Hit:	100.00%	100	–	
Coverpoint ALU_cp	87.50%	100	–	Uncovered
covered/total bins:	7	8	–	
missing/total bins:	1	8	–	
% Hit:	87.50%	100	–	
Cross #cross__0#	100.00%	100	–	Covered
covered/total bins:	8	8	–	
missing/total bins:	0	8	–	
% Hit:	100.00%	100	–	
Cross #cross__1#	100.00%	100	–	Covered
covered/total bins:	18	18	–	
missing/total bins:	0	18	–	
% Hit:	100.00%	100	–	
Cross #cross__2#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Cross #cross__3#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Cross #cross__4#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Cross #cross__5#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Cross #cross__6#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Cross #cross__7#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Cross #cross__8#	76.92%	100	–	Uncovered
covered/total bins:	20	26	–	
missing/total bins:	6	26	–	
% Hit:	76.92%	100	–	
Covergroup instance \/alsu_coverage_pkg::alsu_coverage::cg	98.38%	100	–	Uncovered
covered/total bins:	91	98	–	
missing/total bins:	7	98	–	
% Hit:	92.85%	100	–	
Coverpoint rst_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin auto[0]	947	1	–	Covered
bin auto[1]	53	1	–	Covered
Coverpoint cin_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin auto[0]	493	1	–	Covered
bin auto[1]	507	1	–	Covered
Coverpoint red_op_A_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	

% Hit:	100.00%	100	–	
bin red_op_A_0	185	1	–	Covered
bin red_op_A_1	815	1	–	Covered
default bin red_op_A_default	0		–	ZERO
Coverpoint red_op_B_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin red_op_B_0	155	1	–	Covered
bin red_op_B_1	845	1	–	Covered
default bin red_op_B_default	0		–	ZERO
Coverpoint bypass_A_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin auto[0]	733	1	–	Covered
bin auto[1]	267	1	–	Covered
Coverpoint bypass_B_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin auto[0]	753	1	–	Covered
bin auto[1]	247	1	–	Covered
Coverpoint direction_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin auto[0]	534	1	–	Covered
bin auto[1]	466	1	–	Covered
Coverpoint serial_in_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin auto[0]	521	1	–	Covered
bin auto[1]	479	1	–	Covered
Coverpoint A_cp	100.00%	100	–	Covered
covered/total bins:	5	5	–	
missing/total bins:	0	5	–	
% Hit:	100.00%	100	–	
bin A_data_0	263	1	–	Covered
bin A_data_max	150	1	–	Covered
bin A_data_min	130	1	–	Covered
bin A_data_walkingones[1]	108	1	–	Covered
bin A_data_walkingones[2]	92	1	–	Covered
default bin A_data_default	257		–	Occurred
Coverpoint B_cp	100.00%	100	–	Covered
covered/total bins:	5	5	–	
missing/total bins:	0	5	–	
% Hit:	100.00%	100	–	
bin B_data_0	257	1	–	Covered
bin B_data_max	143	1	–	Covered
bin B_data_min	162	1	–	Covered
bin B_data_walkingones[1]	91	1	–	Covered
bin B_data_walkingones[2]	97	1	–	Covered
default bin B_data_default	250		–	Occurred
Coverpoint A_cp_mod	100.00%	100	–	Covered
covered/total bins:	3	3	–	
missing/total bins:	0	3	–	
% Hit:	100.00%	100	–	
bin A_data_0	263	1	–	Covered
bin A_data_max	150	1	–	Covered
bin A_data_min	130	1	–	Covered
Coverpoint B_cp_mod	100.00%	100	–	Covered
covered/total bins:	3	3	–	
missing/total bins:	0	3	–	
% Hit:	100.00%	100	–	
bin B_data_0	257	1	–	Covered
bin B_data_max	143	1	–	Covered
bin B_data_min	162	1	–	Covered
Coverpoint ALU_cp	87.50%	100	–	Uncovered
covered/total bins:	7	8	–	
missing/total bins:	1	8	–	
% Hit:	87.50%	100	–	
bin Bins_shift[4]	166	1	–	Covered
bin Bins_shift[5]	173	1	–	Covered
bin Bins_arith[2]	188	1	–	Covered
bin Bins_arith[3]	175	1	–	Covered
bin Bins_bitwise[0]	110	1	–	Covered
bin Bins_bitwise[1]	116	1	–	Covered
bin Bins_invalid	72	1	–	Covered
bin Bins_trans	0	1	–	ZERO
Cross #cross__0#	100.00%	100	–	Covered
covered/total bins:	8	8	–	
missing/total bins:	0	8	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin <A_data_walkingones[2],red_op_A_0>	12	1	–	Covered
bin <A_data_walkingones[1],red_op_A_0>	12	1	–	Covered
bin <A_data_min,red_op_A_1>	121	1	–	Covered
bin <A_data_min,red_op_A_0>	9	1	–	Covered
bin <A_data_max,red_op_A_1>	133	1	–	Covered
bin <A_data_0,red_op_A_1>	168	1	–	Covered

bin <A_data_max,red_op_A_0>	17	1	–	Covered
bin <A_data_0,red_op_A_0>	95	1	–	Covered
Illegal and Ignore Bins:				
ignore_bin assert_red_op_A	176		–	Occurred
Cross #cross__1#	100.00%	100	–	Covered
covered/total bins:	18	18	–	
missing/total bins:	0	18	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin <B_data_walkingones[2],red_op_A_1,red_op_B_1>	58	1	–	Covered
bin <B_data_walkingones[2],red_op_A_1,red_op_B_0>	4	1	–	Covered
bin <B_data_walkingones[1],red_op_A_1,red_op_B_1>	62	1	–	Covered
bin <B_data_walkingones[1],red_op_A_1,red_op_B_0>	7	1	–	Covered
bin <B_data_min,red_op_A_1,red_op_B_1>	122	1	–	Covered
bin <B_data_min,red_op_A_1,red_op_B_0>	7	1	–	Covered
bin <B_data_max,red_op_A_1,red_op_B_1>	114	1	–	Covered
bin <B_data_0,red_op_A_1,red_op_B_1>	170	1	–	Covered
bin <B_data_max,red_op_A_1,red_op_B_0>	5	1	–	Covered
bin <B_data_0,red_op_A_1,red_op_B_0>	71	1	–	Covered
bin <B_data_walkingones[2],red_op_A_0,red_op_B_0>	8	1	–	Covered
bin <B_data_walkingones[1],red_op_A_0,red_op_B_0>	2	1	–	Covered
bin <B_data_min,red_op_A_0,red_op_B_1>	26	1	–	Covered
bin <B_data_min,red_op_A_0,red_op_B_0>	7	1	–	Covered
bin <B_data_max,red_op_A_0,red_op_B_1>	15	1	–	Covered
bin <B_data_0,red_op_A_0,red_op_B_1>	12	1	–	Covered
bin <B_data_max,red_op_A_0,red_op_B_0>	9	1	–	Covered
bin <B_data_0,red_op_A_0,red_op_B_0>	4	1	–	Covered
Illegal and Ignore Bins:				
ignore_bin assert_red_op_B	47		–	Occurred
Cross #cross__2#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin arith_permutations	155	1	–	Covered
Cross #cross__3#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin addition_cin	188	1	–	Covered
Cross #cross__4#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin shift_rotate_direction	339	1	–	Covered
Cross #cross__5#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin shift_serial_in	166	1	–	Covered
Cross #cross__6#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin or_xor_red_op_A	22	1	–	Covered
Cross #cross__7#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin or_xor_red_op_B	33	1	–	Covered
Cross #cross__8#	76.92%	100	–	Uncovered
covered/total bins:	20	26	–	
missing/total bins:	6	26	–	
% Hit:	76.92%	100	–	
Auto, Default and User Defined Bins:				
bin <Bins_invalid,red_op_A_1,red_op_B_1>	14	1	–	Covered
bin <Bins_invalid,red_op_A_1,red_op_B_0>	16	1	–	Covered
bin <Bins_invalid,red_op_A_0,red_op_B_1>	21	1	–	Covered
bin <Bins_invalid,red_op_A_0,red_op_B_0>	21	1	–	Covered
bin <Bins_bitwise[1],red_op_A_0,red_op_B_0>	12	1	–	Covered
bin <Bins_bitwise[0],red_op_A_0,red_op_B_0>	14	1	–	Covered
bin <Bins_arith[3],red_op_A_1,red_op_B_1>	160	1	–	Covered
bin <Bins_shift[5],red_op_A_1,red_op_B_1>	162	1	–	Covered

bin <Bins_arith[3],red_op_A_1,red_op_B_0>	4	1	–	Covered
bin <Bins_shift[5],red_op_A_1,red_op_B_0>	5	1	–	Covered
bin <Bins_arith[2],red_op_A_1,red_op_B_1>	172	1	–	Covered
bin <Bins_shift[4],red_op_A_1,red_op_B_1>	145	1	–	Covered
bin <Bins_arith[2],red_op_A_1,red_op_B_0>	9	1	–	Covered
bin <Bins_shift[4],red_op_A_1,red_op_B_0>	8	1	–	Covered
bin <Bins_arith[3],red_op_A_0,red_op_B_1>	10	1	–	Covered
bin <Bins_shift[5],red_op_A_0,red_op_B_1>	6	1	–	Covered
bin <Bins_arith[3],red_op_A_0,red_op_B_0>	1	1	–	Covered
bin <Bins_arith[2],red_op_A_0,red_op_B_1>	7	1	–	Covered
bin <Bins_shift[4],red_op_A_0,red_op_B_1>	12	1	–	Covered
bin <Bins_shift[4],red_op_A_0,red_op_B_0>	1	1	–	Covered
bin <Bins_trans,*,*>	0	1	4	ZERO
bin <Bins_shift[5],red_op_A_0,red_op_B_0>	0	1	1	ZERO
bin <Bins_arith[2],red_op_A_0,red_op_B_0>	0	1	1	ZERO
Illegal and Ignore Bins:				
ignore_bin invalid_reduction	200		–	Occurred

COVERGROUP COVERAGE:

Covergroup	Metric	Goal	Bins	Status
TYPE /alsu_coverage_pkg/alsu_coverage/cg	98.38%	100	–	Uncovered
covered/total bins:	91	98	–	
missing/total bins:	7	98	–	
% Hit:	92.85%	100	–	
Coverpoint rst_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
Coverpoint cin_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
Coverpoint red_op_A_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
Coverpoint red_op_B_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
Coverpoint bypass_A_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
Coverpoint bypass_B_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
Coverpoint direction_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
Coverpoint serial_in_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
Coverpoint A_cp	100.00%	100	–	Covered
covered/total bins:	5	5	–	
missing/total bins:	0	5	–	
% Hit:	100.00%	100	–	
Coverpoint B_cp	100.00%	100	–	Covered
covered/total bins:	5	5	–	
missing/total bins:	0	5	–	
% Hit:	100.00%	100	–	
Coverpoint A_cp_mod	100.00%	100	–	Covered
covered/total bins:	3	3	–	
missing/total bins:	0	3	–	
% Hit:	100.00%	100	–	
Coverpoint B_cp_mod	100.00%	100	–	Covered
covered/total bins:	3	3	–	
missing/total bins:	0	3	–	
% Hit:	100.00%	100	–	
Coverpoint ALU_cp	87.50%	100	–	Uncovered
covered/total bins:	7	8	–	
missing/total bins:	1	8	–	

% Hit:	87.50%	100	–	
Cross #cross__0#	100.00%	100	–	Covered
covered/total bins:	8	8	–	
missing/total bins:	0	8	–	
% Hit:	100.00%	100	–	
Cross #cross__1#	100.00%	100	–	Covered
covered/total bins:	18	18	–	
missing/total bins:	0	18	–	
% Hit:	100.00%	100	–	
Cross #cross__2#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Cross #cross__3#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Cross #cross__4#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Cross #cross__5#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Cross #cross__6#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Cross #cross__7#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Cross #cross__8#	76.92%	100	–	Uncovered
covered/total bins:	20	26	–	
missing/total bins:	6	26	–	
% Hit:	76.92%	100	–	
Covergroup instance \/alsu_coverage_pkg::alsu_coverage::cg	98.38%	100	–	Uncovered
covered/total bins:	91	98	–	
missing/total bins:	7	98	–	
% Hit:	92.85%	100	–	
Coverpoint rst_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin auto[0]	947	1	–	Covered
bin auto[1]	53	1	–	Covered
Coverpoint cin_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin auto[0]	493	1	–	Covered
bin auto[1]	507	1	–	Covered
Coverpoint red_op_A_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin red_op_A_0	185	1	–	Covered
bin red_op_A_1	815	1	–	Covered
default bin red_op_A_default	0		–	ZERO
Coverpoint red_op_B_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin red_op_B_0	155	1	–	Covered
bin red_op_B_1	845	1	–	Covered
default bin red_op_B_default	0		–	ZERO
Coverpoint bypass_A_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin auto[0]	733	1	–	Covered
bin auto[1]	267	1	–	Covered
Coverpoint bypass_B_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin auto[0]	753	1	–	Covered
bin auto[1]	247	1	–	Covered
Coverpoint direction_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin auto[0]	534	1	–	Covered
bin auto[1]	466	1	–	Covered
Coverpoint serial_in_cp	100.00%	100	–	Covered
covered/total bins:	2	2	–	
missing/total bins:	0	2	–	
% Hit:	100.00%	100	–	
bin auto[0]	521	1	–	Covered
bin auto[1]	479	1	–	Covered

Coverpoint A_cp	100.00%	100	–	Covered
covered/total bins:	5	5	–	
missing/total bins:	0	5	–	
% Hit:	100.00%	100	–	
bin A_data_0	263	1	–	Covered
bin A_data_max	150	1	–	Covered
bin A_data_min	130	1	–	Covered
bin A_data_walkingones[1]	108	1	–	Covered
bin A_data_walkingones[2]	92	1	–	Covered
default bin A_data_default	257		–	Occurred
Coverpoint B_cp	100.00%	100	–	Covered
covered/total bins:	5	5	–	
missing/total bins:	0	5	–	
% Hit:	100.00%	100	–	
bin B_data_0	257	1	–	Covered
bin B_data_max	143	1	–	Covered
bin B_data_min	162	1	–	Covered
bin B_data_walkingones[1]	91	1	–	Covered
bin B_data_walkingones[2]	97	1	–	Covered
default bin B_data_default	250		–	Occurred
Coverpoint A_cp_mod	100.00%	100	–	Covered
covered/total bins:	3	3	–	
missing/total bins:	0	3	–	
% Hit:	100.00%	100	–	
bin A_data_0	263	1	–	Covered
bin A_data_max	150	1	–	Covered
bin A_data_min	130	1	–	Covered
Coverpoint B_cp_mod	100.00%	100	–	Covered
covered/total bins:	3	3	–	
missing/total bins:	0	3	–	
% Hit:	100.00%	100	–	
bin B_data_0	257	1	–	Covered
bin B_data_max	143	1	–	Covered
bin B_data_min	162	1	–	Covered
Coverpoint ALU_cp	87.50%	100	–	Uncovered
covered/total bins:	7	8	–	
missing/total bins:	1	8	–	
% Hit:	87.50%	100	–	
bin Bins_shift[4]	166	1	–	Covered
bin Bins_shift[5]	173	1	–	Covered
bin Bins_arith[2]	188	1	–	Covered
bin Bins_arith[3]	175	1	–	Covered
bin Bins_bitwise[0]	110	1	–	Covered
bin Bins_bitwise[1]	116	1	–	Covered
bin Bins_invalid	72	1	–	Covered
bin Bins_trans	0	1	–	ZERO
Cross #cross__0#	100.00%	100	–	Covered
covered/total bins:	8	8	–	
missing/total bins:	0	8	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin <A_data_walkingones[2],red_op_A_0>	12	1	–	Covered
bin <A_data_walkingones[1],red_op_A_0>	12	1	–	Covered
bin <A_data_min,red_op_A_1>	121	1	–	Covered
bin <A_data_min,red_op_A_0>	9	1	–	Covered
bin <A_data_max,red_op_A_1>	133	1	–	Covered
bin <A_data_0,red_op_A_1>	168	1	–	Covered
bin <A_data_max,red_op_A_0>	17	1	–	Covered
bin <A_data_0,red_op_A_0>	95	1	–	Covered
Illegal and Ignore Bins:				
ignore_bin assert_red_op_A	176		–	Occurred
Cross #cross__1#	100.00%	100	–	Covered
covered/total bins:	18	18	–	
missing/total bins:	0	18	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin <B_data_walkingones[2],red_op_A_1,red_op_B_1>	58	1	–	Covered
bin <B_data_walkingones[2],red_op_A_1,red_op_B_0>	4	1	–	Covered
bin <B_data_walkingones[1],red_op_A_1,red_op_B_1>	62	1	–	Covered
bin <B_data_walkingones[1],red_op_A_1,red_op_B_0>	7	1	–	Covered
bin <B_data_min,red_op_A_1,red_op_B_1>	122	1	–	Covered
bin <B_data_min,red_op_A_1,red_op_B_0>	7	1	–	Covered
bin <B_data_max,red_op_A_1,red_op_B_1>	114	1	–	Covered
bin <B_data_0,red_op_A_1,red_op_B_1>	170	1	–	Covered
bin <B_data_max,red_op_A_1,red_op_B_0>	5	1	–	Covered
bin <B_data_0,red_op_A_1,red_op_B_0>	71	1	–	Covered
bin <B_data_walkingones[2],red_op_A_0,red_op_B_0>	8	1	–	Covered
bin <B_data_walkingones[1],red_op_A_0,red_op_B_0>	2	1	–	Covered
bin <B_data_min,red_op_A_0,red_op_B_1>	26	1	–	Covered
bin <B_data_min,red_op_A_0,red_op_B_0>	7	1	–	Covered
bin <B_data_max,red_op_A_0,red_op_B_1>	15	1	–	Covered
bin <B_data_0,red_op_A_0,red_op_B_1>	12	1	–	Covered
bin <B_data_max,red_op_A_0,red_op_B_0>	9	1	–	Covered
bin <B_data_0,red_op_A_0,red_op_B_0>	4	1	–	Covered
Illegal and Ignore Bins:				
ignore_bin assert_red_op_B	47		–	Occurred

Cross #cross__2#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin arith_permutations	155	1	–	Covered
Cross #cross__3#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin addition_cin	188	1	–	Covered
Cross #cross__4#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin shift_rotate_direction	339	1	–	Covered
Cross #cross__5#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin shift_serial_in	166	1	–	Covered
Cross #cross__6#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin or_xor_red_op_A	22	1	–	Covered
Cross #cross__7#	100.00%	100	–	Covered
covered/total bins:	1	1	–	
missing/total bins:	0	1	–	
% Hit:	100.00%	100	–	
Auto, Default and User Defined Bins:				
bin or_xor_red_op_B	33	1	–	Covered
Cross #cross__8#	76.92%	100	–	Uncovered
covered/total bins:	20	26	–	
missing/total bins:	6	26	–	
% Hit:	76.92%	100	–	
Auto, Default and User Defined Bins:				
bin <Bins_invalid,red_op_A_1,red_op_B_1>	14	1	–	Covered
bin <Bins_invalid,red_op_A_1,red_op_B_0>	16	1	–	Covered
bin <Bins_invalid,red_op_A_0,red_op_B_1>	21	1	–	Covered
bin <Bins_invalid,red_op_A_0,red_op_B_0>	21	1	–	Covered
bin <Bins_bitwise[1],red_op_A_0,red_op_B_0>	12	1	–	Covered
bin <Bins_bitwise[0],red_op_A_0,red_op_B_0>	14	1	–	Covered
bin <Bins_arith[3],red_op_A_1,red_op_B_1>	160	1	–	Covered
bin <Bins_shift[5],red_op_A_1,red_op_B_1>	162	1	–	Covered
bin <Bins_arith[3],red_op_A_1,red_op_B_0>	4	1	–	Covered
bin <Bins_shift[5],red_op_A_1,red_op_B_0>	5	1	–	Covered
bin <Bins_arith[2],red_op_A_1,red_op_B_1>	172	1	–	Covered
bin <Bins_shift[4],red_op_A_1,red_op_B_1>	145	1	–	Covered
bin <Bins_arith[2],red_op_A_1,red_op_B_0>	9	1	–	Covered
bin <Bins_shift[4],red_op_A_1,red_op_B_0>	8	1	–	Covered
bin <Bins_arith[3],red_op_A_0,red_op_B_1>	10	1	–	Covered
bin <Bins_shift[5],red_op_A_0,red_op_B_1>	6	1	–	Covered
bin <Bins_arith[3],red_op_A_0,red_op_B_0>	1	1	–	Covered
bin <Bins_arith[2],red_op_A_0,red_op_B_1>	7	1	–	Covered
bin <Bins_shift[4],red_op_A_0,red_op_B_1>	12	1	–	Covered
bin <Bins_shift[4],red_op_A_0,red_op_B_0>	1	1	–	Covered
bin <Bins_trans,*,*>	0	1	4	ZERO
bin <Bins_shift[5],red_op_A_0,red_op_B_0>	0	1	1	ZERO
bin <Bins_arith[2],red_op_A_0,red_op_B_0>	0	1	1	ZERO
Illegal and Ignore Bins:				
ignore_bin invalid_reduction	200		–	Occurred

TOTAL COVERGROUP COVERAGE: 98.38% COVERGROUP TYPES: 1

Total Coverage By Instance (filtered view): 98.38%

1.5 5. code Coverage Report

```
Coverage Report by DU with details

=================================================================================
=== Design Unit: work.ALSU
=================================================================================
Branch Coverage:
    Enabled Coverage                 Bins      Hits    Misses  Coverage
    ----------------                 ----      ----    ------  --------
    Branches                           32        32         0   100.00%

================================Branch Details================================

Branch Coverage for Design Unit work.ALSU

    Line         Item                           Count     Source
    ----         ----                           -----     ------
  File ALSU.sv
------------------------------------IF Branch------------------------------------
    24                                          1048     Count coming in to IF
    24              1                            101       if(rst) begin
    35              1                            947       end else begin
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    51                                          1048     Count coming in to IF
    51              1                            101       if(rst) begin
    53              1                            947       end else begin
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    54                                           947     Count coming in to IF
    54              1                            690           if (invalid)
    56              1                            257           else
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    63                                          1048     Count coming in to IF
    63              1                            101       if(rst) begin
    66              1                            947       else begin
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    67                                           947     Count coming in to IF
    67              1                             59         if (bypass_A_reg && bypass_B_reg)
    69              1                            176         else if (bypass_A_reg)
    71              1                            164         else if (bypass_B_reg)
    73              1                            387         else if (invalid)
    75              1                            161         else begin
Branch totals: 5 hits of 5 branches = 100.00%

-----------------------------------CASE Branch-----------------------------------
    76                                           161     Count coming in to CASE
    77              1                             24               3'h0: begin
    87              1                             19               3'h1: begin
    97              1                             29               3'h2: out <= A_reg + B_reg;
    98              1                             24               3'h3: out <= A_reg * B_reg;
    99              1                             31               3'h4: begin
    105             1                             27               3'h5: begin
                                                   7     All False Count
Branch totals: 7 hits of 7 branches = 100.00%

------------------------------------IF Branch------------------------------------
    78                                            24     Count coming in to IF
    78              1                              5               if (red_op_A_reg && red_op_B_reg)
    80              1                              5               else if (red_op_A_reg)
    82              1                              4               else if (red_op_B_reg)
    84              1                             10               else
Branch totals: 4 hits of 4 branches = 100.00%

------------------------------------IF Branch------------------------------------
    88                                            19     Count coming in to IF
    88              1                              2               if (red_op_A_reg && red_op_B_reg)
    90              1                              5               else if (red_op_A_reg)
    92              1                              5               else if (red_op_B_reg)
    94              1                              7               else
Branch totals: 4 hits of 4 branches = 100.00%

------------------------------------IF Branch------------------------------------
    100                                           31     Count coming in to IF
    100             1                             11               if (direction_reg)
    102             1                             20               else
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    106                                           27     Count coming in to IF
    106             1                              8               if (direction_reg)
    108             1                             19               else
Branch totals: 2 hits of 2 branches = 100.00%
```

Condition Coverage:

Enabled Coverage	Bins	Covered	Misses	Coverage
Conditions	6	6	0	100.00%

==========Condition Details==========

Condition Coverage for Design Unit work.ALSU —

File ALSU.sv

----------Focused Condition View----------

Line 67 Item 1 (bypass_A_reg && bypass_B_reg)
Condition totals: 2 of 2 input terms covered = 100.00%

Input Term	Covered	Reason for no coverage	Hint
bypass_A_reg	Y		
bypass_B_reg	Y		

Rows:	Hits	FEC Target	Non-masking condition(s)
Row 1:	1	bypass_A_reg_0	-
Row 2:	1	bypass_A_reg_1	bypass_B_reg
Row 3:	1	bypass_B_reg_0	bypass_A_reg
Row 4:	1	bypass_B_reg_1	bypass_A_reg

----------Focused Condition View----------

Line 78 Item 1 (red_op_A_reg && red_op_B_reg)
Condition totals: 2 of 2 input terms covered = 100.00%

Input Term	Covered	Reason for no coverage	Hint
red_op_A_reg	Y		
red_op_B_reg	Y		

Rows:	Hits	FEC Target	Non-masking condition(s)
Row 1:	1	red_op_A_reg_0	-
Row 2:	1	red_op_A_reg_1	red_op_B_reg
Row 3:	1	red_op_B_reg_0	red_op_A_reg
Row 4:	1	red_op_B_reg_1	red_op_A_reg

----------Focused Condition View----------

Line 88 Item 1 (red_op_A_reg && red_op_B_reg)
Condition totals: 2 of 2 input terms covered = 100.00%

Input Term	Covered	Reason for no coverage	Hint
red_op_A_reg	Y		
red_op_B_reg	Y		

Rows:	Hits	FEC Target	Non-masking condition(s)
Row 1:	1	red_op_A_reg_0	-
Row 2:	1	red_op_A_reg_1	red_op_B_reg
Row 3:	1	red_op_B_reg_0	red_op_A_reg
Row 4:	1	red_op_B_reg_1	red_op_A_reg

Expression Coverage:

Enabled Coverage	Bins	Covered	Misses	Coverage
Expressions	8	8	0	100.00%

==========Expression Details==========

Expression Coverage for Design Unit work.ALSU —

File ALSU.sv

----------Focused Expression View----------

Line 18 Item 1 ((red_op_A_reg | red_op_B_reg) & (opcode_reg[1] | opcode_reg[2]))
Expression totals: 4 of 4 input terms covered = 100.00%

Input Term	Covered	Reason for no coverage	Hint
red_op_A_reg	Y		
red_op_B_reg	Y		
opcode_reg[1]	Y		
opcode_reg[2]	Y		

Rows:	Hits	FEC Target	Non-masking condition(s)
Row 1:	1	red_op_A_reg_0	((opcode_reg[1] | opcode_reg[2]) && ~red_op_B_reg)
Row 2:	1	red_op_A_reg_1	((opcode_reg[1] | opcode_reg[2]) && ~red_op_B_reg)
Row 3:	1	red_op_B_reg_0	((opcode_reg[1] | opcode_reg[2]) && ~red_op_A_reg)
Row 4:	1	red_op_B_reg_1	((opcode_reg[1] | opcode_reg[2]) && ~red_op_A_reg)
Row 5:	1	opcode_reg[1]_0	((red_op_A_reg | red_op_B_reg) && ~opcode_reg[2])
Row 6:	1	opcode_reg[1]_1	((red_op_A_reg | red_op_B_reg) && ~opcode_reg[2])
Row 7:	1	opcode_reg[2]_0	((red_op_A_reg | red_op_B_reg) && ~opcode_reg[1])
Row 8:	1	opcode_reg[2]_1	((red_op_A_reg | red_op_B_reg) && ~opcode_reg[1])

```
----------Focused Expression View----------
Line       19 Item    1  (opcode_reg[1] & opcode_reg[2])
Expression totals: 2 of 2 input terms covered = 100.00%

     Input Term   Covered  Reason for no coverage   Hint
     ----------   -------  ----------------------   ----
  opcode_reg[1]         Y
  opcode_reg[2]         Y

    Rows:       Hits  FEC Target            Non-masking condition(s)
    -----       ----  ----------            ------------------------
  Row    1:        1  opcode_reg[1]_0       opcode_reg[2]
  Row    2:        1  opcode_reg[1]_1       opcode_reg[2]
  Row    3:        1  opcode_reg[2]_0       opcode_reg[1]
  Row    4:        1  opcode_reg[2]_1       opcode_reg[1]

----------Focused Expression View----------
Line       20 Item    1  (invalid_red_op | invalid_opcode)
Expression totals: 2 of 2 input terms covered = 100.00%

     Input Term   Covered  Reason for no coverage   Hint
     ----------   -------  ----------------------   ----
  invalid_red_op        Y
  invalid_opcode        Y

    Rows:       Hits  FEC Target            Non-masking condition(s)
    -----       ----  ----------            ------------------------
  Row    1:        1  invalid_red_op_0      ~invalid_opcode
  Row    2:        1  invalid_red_op_1      ~invalid_opcode
  Row    3:        1  invalid_opcode_0      ~invalid_red_op
  Row    4:        1  invalid_opcode_1      ~invalid_red_op

Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                      48        48         0   100.00%

================================Statement Details================================

Statement Coverage for Design Unit work.ALSU —

    Line         Item                    Count     Source
    ----         ----                    -----     ------
  File ALSU.sv
       1                                           module ALSU(A, B, cin, serial_in, red_op_A, red_op_B, opcode, bypass_A, bypass_B, c
       2                                           parameter INPUT_PRIORITY = "A";
       3                                           parameter FULL_ADDER = "ON";
       4                                           input clk, cin, rst, red_op_A, red_op_B, bypass_A, bypass_B, direction, serial_in;
       5                                           input [2:0] opcode;
       6                                           input signed [2:0] A, B;
       7                                           output reg [15:0] leds;
       8                                           output reg signed [5:0] out;
       9
      10                                           reg red_op_A_reg, red_op_B_reg, bypass_A_reg, bypass_B_reg, direction_reg, serial_in
      11                                           reg signed cin_reg;
      12                                           reg [2:0] opcode_reg;
      13                                           reg signed [2:0] A_reg, B_reg;
      14
      15                                           wire invalid_red_op, invalid_opcode, invalid;
      16
      17                                           //Invalid handling
      18          1                         889    assign invalid_red_op = (red_op_A_reg | red_op_B_reg) & (opcode_reg[1] | opcode_reg
      19          1                         848    assign invalid_opcode = opcode_reg[1] & opcode_reg[2];
      20          1                         505    assign invalid = invalid_red_op | invalid_opcode;
      21
      22                                           //Registering input signals
      23          1                        1048    always @(posedge clk or posedge rst) begin
      24                                             if(rst) begin
      25          1                         101        cin_reg <= 0;
      26          1                         101        red_op_B_reg <= 0;
      27          1                         101        red_op_A_reg <= 0;
      28          1                         101        bypass_B_reg <= 0;
      29          1                         101        bypass_A_reg <= 0;
      30          1                         101        direction_reg <= 0;
      31          1                         101        serial_in_reg <= 0;
      32          1                         101        opcode_reg <= 0;
      33          1                         101        A_reg <= 0;
      34          1                         101        B_reg <= 0;
      35                                             end else begin
      36          1                         947        cin_reg <= cin;
      37          1                         947        red_op_B_reg <= red_op_B;
      38          1                         947        red_op_A_reg <= red_op_A;
      39          1                         947        bypass_B_reg <= bypass_B;
      40          1                         947        bypass_A_reg <= bypass_A;
      41          1                         947        direction_reg <= direction;
      42          1                         947        serial_in_reg <= serial_in;
      43          1                         947        opcode_reg <= opcode;
      44          1                         947        A_reg <= A;
      45          1                         947        B_reg <= B;
      46                                             end
      47                                           end
```

```
48
49                                      //leds output blinking
50              1                1048   always @(posedge clk or posedge rst) begin
51                                        if(rst) begin
52              1                 101          leds <= 0;
53                                        end else begin
54                                            if (invalid)
55              1                 690            leds <= ~leds;
56                                            else
57              1                 257            leds <= 0;
58                                        end
59                                      end
60
61                                      //ALSU output processing
62              1                1048   always @(posedge clk or posedge rst) begin
63                                        if(rst) begin
64              1                 101        out <= 0;
65                                        end
66                                        else begin
67                                          if (bypass_A_reg && bypass_B_reg)
68              1                  59           out <= (INPUT_PRIORITY == "A")? A_reg: B_reg;
69                                          else if (bypass_A_reg)
70              1                 176           out <= A_reg;
71                                          else if (bypass_B_reg)
72              1                 164           out <= B_reg;
73                                          else if (invalid)
74              1                 387              out <= 0;
75                                          else begin
76                                              case (opcode)
77                                                3'h0: begin
78                                                  if (red_op_A_reg && red_op_B_reg)
79              1                   5                 out <= (INPUT_PRIORITY == "A")? |A_reg: |B_reg;
80                                                else if (red_op_A_reg)
81              1                   5                 out <= |A_reg;
82                                                else if (red_op_B_reg)
83              1                   4                 out <= |B_reg;
84                                                else
85              1                  10                 out <= A_reg | B_reg;
86                                              end
87                                              3'h1: begin
88                                                if (red_op_A_reg && red_op_B_reg)
89              1                   2                 out <= (INPUT_PRIORITY == "A")? ^A_reg: ^B_reg;
90                                                else if (red_op_A_reg)
91              1                   5                 out <= ^A_reg;
92                                                else if (red_op_B_reg)
93              1                   5                 out <= ^B_reg;
94                                                else
95              1                   7                 out <= A_reg ^ B_reg;
96                                              end
97              1                  29              3'h2: out <= A_reg + B_reg;
98              1                  24              3'h3: out <= A_reg * B_reg;
99                                              3'h4: begin
100                                               if (direction_reg)
101             1                  11               out <= {out[4:0], serial_in_reg};
102                                             else
103             1                  20               out <= {serial_in_reg, out[5:1]};
104                                             end
105                                             3'h5: begin
106                                               if (direction_reg)
107             1                   8               out <= {out[4:0], out[5]};
108                                             else
109             1                  19               out <= {out[0], out[5:1]};
```

Toggle Coverage:

Enabled Coverage	Bins	Hits	Misses	Coverage
Toggles	118	118	0	100.00%

Toggle Details

Toggle Coverage for Design Unit work.ALSU

Node	1H–>0L	0L–>1H	"Coverage"
A[0–2]	1	1	100.00
A_reg[0–2]	1	1	100.00
B[0–2]	1	1	100.00
B_reg[0–2]	1	1	100.00
bypass_A	1	1	100.00
bypass_A_reg	1	1	100.00
bypass_B	1	1	100.00
bypass_B_reg	1	1	100.00
cin	1	1	100.00
cin_reg	1	1	100.00
clk	1	1	100.00
direction	1	1	100.00
direction_reg	1	1	100.00
invalid	1	1	100.00
invalid_opcode	1	1	100.00
invalid_red_op	1	1	100.00
leds[0–15]	1	1	100.00
opcode[0–2]	1	1	100.00

opcode_reg[0-2]	1	1	100.00
out[0-5]	1	1	100.00
red_op_A	1	1	100.00
red_op_A_reg	1	1	100.00
red_op_B	1	1	100.00
red_op_B_reg	1	1	100.00
rst	1	1	100.00
serial_in	1	1	100.00
serial_in_reg	1	1	100.00

```
Total Node Count        =        59
Toggled Node Count      =        59
Untoggled Node Count    =         0

Toggle Coverage         =    100.00% (118 of 118 bins)


Total Coverage By Design Unit (filtered view): 100.00%
```

1.6 6. Waveform

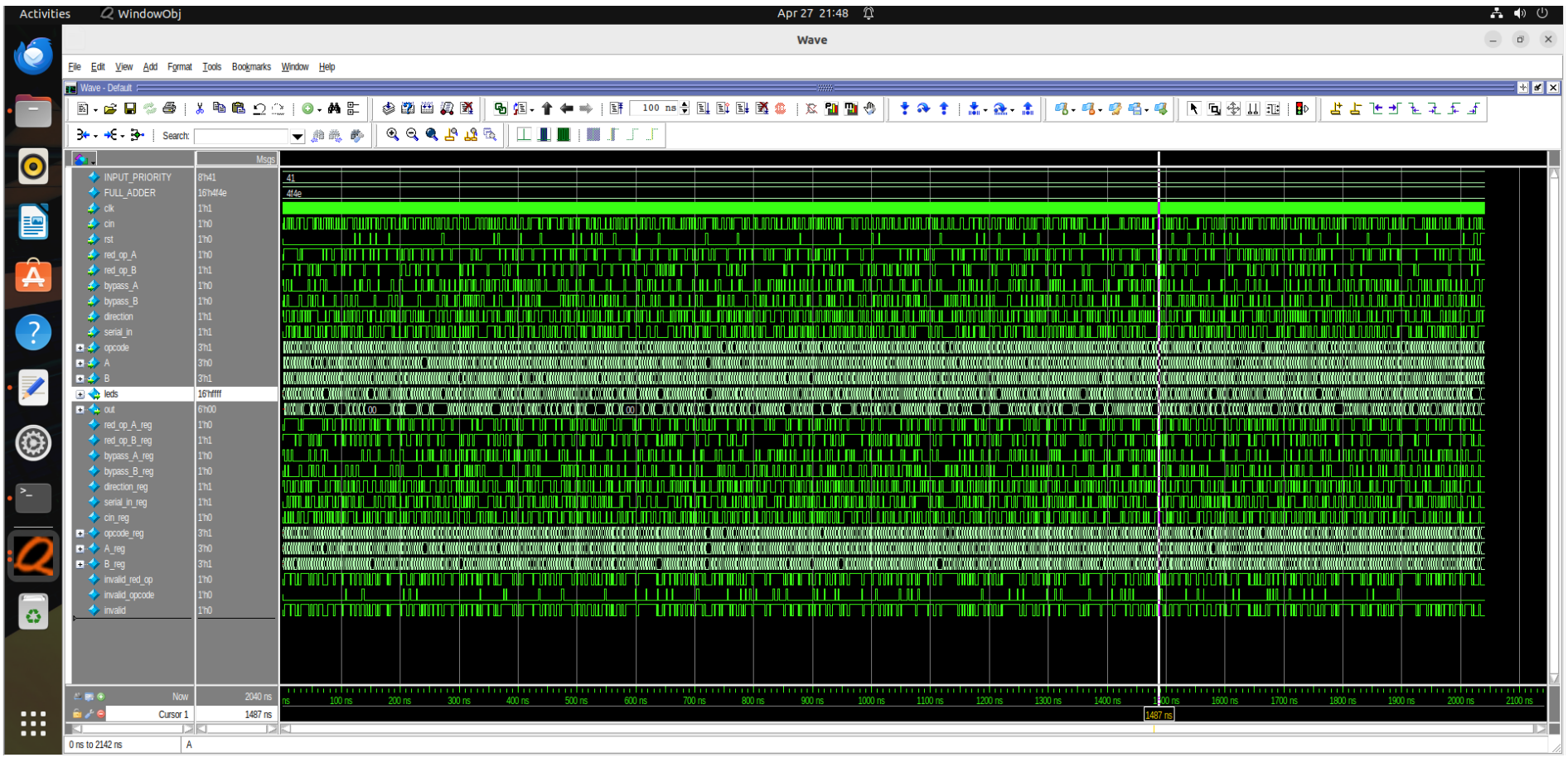

Figure 1: simulation waveform

```
# (C) 2007-2013 Cadence Design Systems, Inc.
# (C) 2006-2013 Synopsys, Inc.
# (C) 2011-2013 Cypress Semiconductor Corp.
# ------------------------------------------------------------
#
#   ***********       IMPORTANT RELEASE NOTES         ************
#
#   You are using a version of the UVM library that has been compiled
#   with `UVM_NO_DEPRECATED undefined.
#   See http://www.eda.org/svdb/view.php?id=3313 for more details.
#
#   You are using a version of the UVM library that has been compiled
#   with `UVM_OBJECT_MUST_HAVE_CONSTRUCTOR undefined.
#   See http://www.eda.org/svdb/view.php?id=3770 for more details.
#
#       (Specify +UVM_NO_RELNOTES to turn off this notice)
#
# UVM_INFO verilog_src/questa_uvm_pkg-1.2/src/questa_uvm_pkg.sv(277) @ 0: reporter [Questa UVM] QUESTA_UVM-1.2.3
# UVM_INFO verilog_src/questa_uvm_pkg-1.2/src/questa_uvm_pkg.sv(278) @ 0: reporter [Questa UVM]  questa_uvm::init(all)
# UVM_INFO @ 0: reporter [RNTST] Running test alsu_test...
# ** Warning: (vsim-8474) A higher value '4' is found in bin 'A_data_walkingones' of Coverpoint 'A_cp'. It is invalid and will be ignored.
#    Time: 0 ns  Iteration: 7  Process: /uvm_pkg::uvm_phase::m_run_phases/#FORK#1847_7feff0488f8 File: alsu_coverage_pkg.sv Line: 52
# ** Warning: (vsim-8474) A higher value '4' is found in bin 'B_data_walkingones' of Coverpoint 'B_cp'. It is invalid and will be ignored.
#    Time: 0 ns  Iteration: 7  Process: /uvm_pkg::uvm_phase::m_run_phases/#FORK#1847_7feff0488f8 File: alsu_coverage_pkg.sv Line: 59
# UVM_INFO alsu_test_pkg.sv(35) @ 0: uvm_test_top [run_phase] Inside the ALSU test.
# ****************************************************************************
# * Questa UVM Transaction Recording Turned ON.                              *
# * recording_detail has been set.                                           *
# *  To turn off, set 'recording_detail' to off:                             *
# * uvm_config_db#(int)            ::set(null, "", "recording_detail", 0); *
# * uvm_config_db#(uvm_bitstream_t)::set(null, "", "recording_detail", 0); *
# ****************************************************************************
# UVM_INFO verilog_src/uvm-1.1d/src/base/uvm_objection.svh(1267) @ 2040: reporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
# UVM_INFO alsu_scoreboard_pkg.sv(130) @ 2040: uvm_test_top.env.sb [REPORT] Total correct transactions: 1020
# UVM_INFO alsu_scoreboard_pkg.sv(131) @ 2040: uvm_test_top.env.sb [REPORT] Total mismatches: 0
#
# --- UVM Report Summary ---
#
# ** Report counts by severity
# UVM_INFO :    7
# UVM_WARNING :    0
# UVM_ERROR :    0
# UVM_FATAL :    0
# ** Report counts by id
# [Questa UVM]     2
# [REPORT]     2
# [RNTST]     1
# [TEST_DONE]     1
# [run_phase]     1
# ** Note: $finish    : /usr/local/questasim/linux_x86_64/../verilog_src/uvm-1.1d/src/base/uvm_root.svh(430)
#    Time: 2040 ns  Iteration: 61  Instance: /top
```

Figure 2: Transcript : all test cases passed