1. Verification requirement document

Label	Description	Stimulus Generation	Functional Coverage	Functionality Check
ALSU_1	the addition on ports A and B taking cin if parameter FULL_ADDER is high	constraints on A and B to have the maximum, minimum and zero values	Included as coverpoint for A and B. Included with cross coverage when ALU opcode is addition with cin taking 0 or 1	Output checked against golden model
ALSU_2	the multiplication on ports A and B	constraints on A and B to have the maximum, minimum and zero values	Included in coverpoint for A and B. Included with cross coverage when ALU opcode is multiplication	Output checked against golden model
ALSU_3	B. incase of red_op inputs are high, then out should be the reduction operation on port A or B based on the priority if the both red_op ports are high.	constraints on A and B, incase of red_op_A is high, constrain A to have one bit high most of the time and B to be zero.	B. Included with cross coverage when ALU opcode is OR or	Output checked against golden model
ALSU_4	When invalid cases exist (opcode is 110/111 or red_op inputs are high when opcode isn't OR/XOR), out should be low and leds should blink	Randomization under constraints where invalid cases do not occur as frequent as valid cases	Included in a coverpoint for opcode. Included with cross coverage to make sure invalid cases occur	Output checked against golden model
ALSU_5	If invalid cases do not occur and the bypass inputs are high, then the output out should by bypass port A or B based on the prioirty if the both bypass ports are high		Included in a coverpoint for bypass	Output checked against golden model
ALSU_6	Incase of invalid cases do not occur, when opcode is SHIFT, then out should be shifted left or right with adding "serial_in" input based on "direction" input is high or low respectively.	Randomization for direction and serial_in inputs with no constraints on both of them and on ports A and B.	opcode.	Output checked against golden model
ALSU_7	Incase of invalid cases do not occur, when opcode is ROTATE, then out should be rotated left or right based on "direction" input is high or low respectively.	Randomization for direction input with no constraints on it and on ports A and B.	Included in a coverpoint for opcode. Included in cross coverage with direction taking 0 or 1	Output checked against golden model
ALSU_8	When the asynchronous reset is asserted, then out and leds output should be low.	Randomization for rst input to be asserted with a low probability.	not included	Output Checked against golden model

2. Code Coverage Report

```
______
=== Instance: /ALSU tb/dut
=== Design Unit: work.ALSU
______
Branch Coverage:
                   Bins Hits Misses Coverage
  Enabled Coverage
                           27 4 coverage
  -----
                     ----
                          ----
                      28
  Branches
Branch Coverage for instance /ALSU_tb/dut
                                Source
                          Count
  Line
          Item
Statement Coverage:
 Enabled Coverage
                 Bins Hits Misses Coverage
                  ----
                       ----
  Statements
                   46
                        46
                             0 100.00%
Statement Coverage for instance /ALSU_tb/dut --
        Item
  Line
                       Count Source
                       ----
        ----
                            -----
Expression Coverage:
 Enabled Coverage
                Bins Covered Misses Coverage
                      coverage
                           0 100.00%
 Expressions
Expression Coverage for instance /ALSU_tb/dut --
File ALSU.sv
-----Focused Expression View------
     24 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%
```

Condition Coverage:					
Enabled Coverage	Bins	Covere	ed M	isses	Coverage
Conditions	6		6	0	100.00%
	=====Conditi	ion Detai	ls====		
Condition Coverage for in	stance /ALSU_t	b/dut	20		
File ALSU.sv					
Focused Co	andition View				
Line 60 Item 1			The state of the s		
Condition totals: 2 of 2	input terms co	overea =	100.00	/6	
Toggle Coverage:					
Enabled Coverage	Bins	Hits	Misses	Cover	age
	118	118	0	100.	00%
Toggles	110				
13.31	13//2015/12/09/10				
Toggles	13//2015/12/09/10	ails====			
13.31	====Toggle Deta				

Name		_	Design UnitType	Lang	File(Line)	Hits Status
The state of the s	k_result/leds_check_cover	Control of the contro				
	k_result/out_check_cover nc_reset_cover				ALSU_tb.sv(211)	
TOTAL DIRECT	VE COVERAGE: 100.00% COVE	RS: 3				
ASSERTION RES	SULTS:					
Name	File(Line)		Failure Count		Count	
/ALSU tb/#ub]	k#102282978#43/immed 44					
	ALSU_tb.sv(44)		(9	1	
/ALSU_tb/#ub]	k#102282978#65/immed67					
	ALSU_tb.sv(67)		(3	1	
/ALSU_tb/#and	onb1k#102282978#86#4#/#ub1k	#10228297	78#86/imme	ed_ 87	7	
	ALSU_tb.sv(87)		(9	1	
/ALSU_tb/ched	k_result/leds_check_assert					
	ALSU_tb.sv(200)		(9	1	
/ALSU_tb/chec	k_result/out_check_assert					
	ALSU_tb.sv(202)		(9	1	
/ALSII +b /Acur	c_reset_assert					
/ALSU_LU/ASYI	ALSU tb.sv(210)		1	3	1	

3. Functional Coverage Report

Covergroup Coverage:	4		100	0.00/		
Covergroups	1	na	na 100.	100000		
Coverpoints/Crosses	17	na	na 0 100	na		
Covergroup Bins	31	31	0 100.	. 00%		
overgroup			Metric	Goal	Bins	Status
Coverpoint A_cp			100.00%	100	(i)	Covered
covered/total bins:			3	3	0.40	
missing/total bins:			0	3	10.00	
% Hit:			100.00%	100	-	
bin A_data_0			8990	1	-	Covered
bin A_data_max			9067	1	((4))	Covered
bin A_data_min			8963	1	15 <u>18 18 18 18 18 18 18 18 18 18 18 18 18 1</u>	Covered
default bin A_data_defau	ult		41540		-	Occurred
Coverpoint A_walkingones_cp			100.00%	100		Covered
covered/total bins:			3	3	92 <u>4</u> 9	
missing/total bins:			0	3	10.00	
% Hit:			100.00%	100		
bin A_data_walkingones[-4]		726	1		Covered
bin A_data_walkingones[1]		487	1	929	Covered
bin A_data_walkingones[2	2]		477	1	15 <u>7</u> 0	Covered
Coverpoint B_cp			100.00%	100	1 - 1	Covered
covered/total bins:			3	3	-	
missing/total bins:			0	3	8548	
% Hit:			100.00%	100	12 <u>0</u> 0	
bin B_data_0			9559	1	-	Covered
bin B_data_max			9154	1	-	Covered
bin B_data_min			9057	1	9548	Covered
default bin B_data_defau	ult		40790		120	Occurred
Coverpoint B_walkingones_cp			100.00%	100	1.75	Covered
covered/total bins:			3	3	-	
missing/total bins:			0	3	948	
% Hit:			100.00%	100	_	
bin B_data_walkingones[-4]		361	1	1575	Covered
bin B_data_walkingones[:	1]		208	1	-	Covered
bin B data walkingones[2]		207	1	(4)	Covered

Coverpoint ALU_cp	100.00%	100	둰	Covered
covered/total bins:	7	7	=	
missing/total bins:	0	7	28	
% Hit:	100.00%	100	70	
illegal_bin Bins_invalid	15976		-	Occurred
bin Bins_shift[SHIFT]	9080	1	49	Covered
bin Bins_shift[ROTATE]	8310	1	2	Covered
bin Bins_arith[ADD]	8817	1	7.	Covered
bin Bins arith[MULT]	8953	1	-	Covered
bin Bins bitwise[OR]	8932	1	49	Covered
bin Bins_bitwise[XOR]	8492	1	2	Covered
bin Bins_trans	1	1	7.0	Covered
Coverpoint cin_cp [1]	100.00%	100	-	Covered
covered/total bins:	1	1	40	
missing/total bins:	0	1	2	
% Hit:	100.00%	100	-	
bin cin0_1	68560	1	-	Covered
Coverpoint serial_in_cp [1]	100.00%	100	49	Covered
covered/total bins:	1	1	2	
missing/total bins:	0	1	70	
% Hit:	100.00%	100	-	
bin serial_in0_1	68560	1	20	Covered
Coverpoint direction_cp [1]	100.00%	100	2	Covered
covered/total bins:	1	1	50	
missing/total bins:	0	1	7.5	
% Hit:	100.00%	100	49	
bin direction0_1	68560	1	2	Covered
Coverpoint red_op_A_cp [1]	100.00%	100	70	Covered
covered/total bins:	1	1	===	
missing/total bins:	0	1	4	
% Hit:	100.00%	100	2	
bin red op A high	3619	1	-	Covered

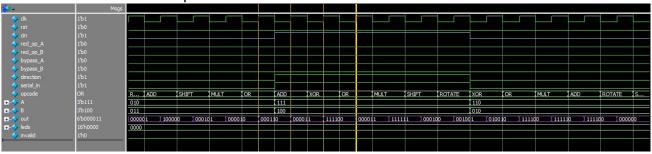
Coverpoint red_op_B_cp [1]	100.00%	100		Covered
covered/total bins:	1	1	=	
missing/total bins:	0	1	<u> 2</u>	
% Hit:	100.00%	100	. 	
bin red_op_B_high	3605	1	-	Covered
Cross arith_corner_cases_cross	100.00%	100	2	Covered
covered/total bins:	1	1	2	
missing/total bins:	0	1	_	
% Hit:	100.00%	100	-	
Auto, Default and User Defined Bins:	100.00%	100		
bin arith_corner_cases	3492	1	- 33	Covered
	100.00%	100	-	Covered
Cross ADD_cin_cross			177	covered
covered/total bins:	1	1		
missing/total bins:	0	1		
% Hit:	100.00%	100	_	
Auto, Default and User Defined Bins:				
bin ADD_cin0_1	8817	1	17	Covered
Cross SHIFT_serial_in_cross	100.00%	100	*	Covered
covered/total bins:	1	1	2	
missing/total bins:	0	1	177	
% Hit:	100.00%	100	17	
Auto, Default and User Defined Bins:				
bin SHIFT_serial_in0_1	9080	1	<u> 2</u>	Covered
Cross SHIFT direction cross	100.00%	100	17	Covered
covered/total bins:	1	1	-	
missing/total bins:	0	1		
% Hit:	100.00%	100	22	
Auto, Default and User Defined Bins:	100.00%	100		
bin SHIFT_ROTATE_direction0_1	17390	1	_	Covered
Cross bitwise_red_op_A_cross	100.00%	100		Covered
[2, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15			-	covered
covered/total bins:	1	1	-	
missing/total bins:	0	1	177	
% Hit:	100.00%	100	177	
Auto, Default and User Defined Bins:				
bin bitwise_A_pattern	584	1	2	Covered
Cross bitwise_red_op_B_cross	100.00%	100	1,5	Covered
covered/total bins:	1	1		
missing/total bins:	0	1	32	
% Hit:	100.00%	100		
Auto, Default and User Defined Bins:				
bin bitwise_B_pattern	247	1		Covered
Cross Invalid_case_cross	100.00%	100	- 2	Covered
covered/total bins:	1	1	_	covered
missing/total bins:	0	1	- ST	
% Hit:	100.00%	100	-	
Auto, Default and User Defined Bins:	4077	4		· .
bin invalid_case	1277	1	- 17	Covered

4. Waveforms

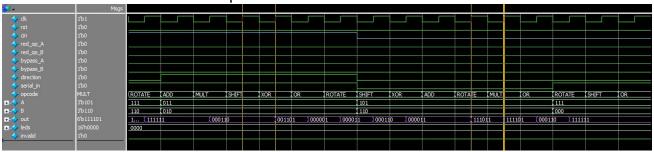
✓ ADD and MULT opcodes results

€1+		Msgs																				
	dk	1'b1																				
-	rst	1'b0																	10.00			
	cin	1'b1	_												1							
	red_op_A	1'b0		 													l			ļ		
	red_op_B	1'b0																				
4	bypass_A	1'b0																				
4	bypass_B	1'b0	5																			
-	direction	1'b1													1							
	serial_in	1'b1																				
4	opcode	SHIFT	(MULT	OR	ADD	Ixo	OR	OR		MULT		SHIFT		ROTATE	XOR	OR	MULT	ADD	ROTATE	SHIFT	MULT	ADD
₽-♦	A	-3'd1	2		-1										-2						-1	
⊞-4	В	-3'd4	3		[-4										2						-2	
	out	6'd4	5 2	(6		3	-4		3		-1		4	9	18	-4	/-2	\ -4	(0), 2
±-4		16'h0000	0000	i i		- 2																
-	invalid	1'h0	2																			

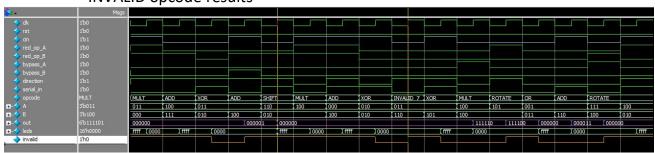
✓ OR and XOR opcodes results



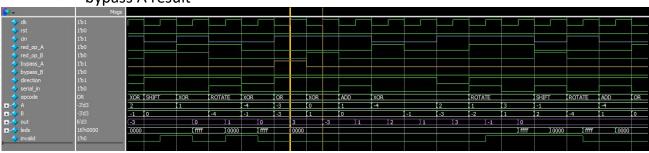
✓ SHIFT and ROTATE opcodes results



✓ INVALID opcode results



✓ bypass A result



✓ INPUT_PRIORITY (A) check

& i •	Msgs			_		2													
♦ dk	1'b1																		
√ rst	1'b0																		
💠 cin	1'b1									,									
red_op_A	1b1																		
red_op_B	1b1									1									
bypass_A	1'b0																		
bypass_B	1'b0				_														
direction	1b1									73									
serial_in	1'b1									80									
opcode	MULT	SHIFT	XOR	ADD	XOR	1)	MULT		ADD	6	OR	INVALID 7	ROTATE	SHIFT	OR		INVALID 7	XOR	ADD
 A	3'b010	011	000	011	100	(10		011	100	010		001	011	100	011		001	000
 -♦ B	3'b011	010	100		[000)11				111	I 000		[001	111	101		000	010
- → out	6'b000001	0 111				000000		00000				(1111				(1111			
≖	16'h0000	ffff 000	io (ffff	(0000		ffff		0000	ffff	(0000	(ffff	(0000	(ffff	(0000	(ffff	0000		X fffff	(0000
🔷 invalid	1'h1																		
8																			

✓ Constraint no. 8 (array of valid unique opcodes)

△	Msgs																
→ dk	1'b0																
→ rst	1'b0																
🥠 cin	1'b1																
red_op_A	1'b0																
red_op_B	1'b0																
bypass_A	1'b0																
bypass_B	1'b0																
direction	1'b1																
serial_in	1'b1																
opcode		XOR	MULT	SHIFT	MULT	XOR	ADD	ROTATE	OR	ROTATE	MULT	ADD	SHIFT	XOR	OR		MULT
⊞♦ A		000		001						110						111	
■ -◆ B		100		111												110	
 → out	6'b111100	1 11100	1 (11110	00000	00000	1 11111	1 (1111)	00000	1 (0000	0 (1111	1	(00001	0 11110	1 (0111	00000	1 (11111	1
		0000															
🥠 invalid	1'h0																
		0		22			Sic										