Verichip's ALU Test Plan

Created by Akshat Baranwal

1. Verichip in **Normal** State (Section 6.2, 5.1, 5.2, 5.3, 5.5, 5.6, 5.7)

   1.1 Objective:

   - 1.1.1 Verify arithmetic operations, logical shifts and data movement.
   - 1.1.2 Verify state transitions are triggered by valid/invalid commands or data conditions.

   1.2 Test Table - Valid (Command Register) is 1'b1:

   - 1.2.1 Verify whether valid bit self clears for each test case.
   - 1.2.2 Unless specified explicitly in Expected Result column, Status Register is 0x0101(Normal State)
   - 1.2.3 The signal export_disable is 1'b0 (Version Register = 0x0210), unless specified.
   - 1.2.4 Test name with multiple test cases use a single value of register either left_alu_reg or right_alu_reg with multiple value for right_alu_reg or left_alu_reg values respectively.

Test name	Command Register	ALU operation	Test Description	Expected Result
Alu_normal_no_cmd	0x8000	No command	left_alu_reg = 0x0001 right_alu_reg = 0x0002 alu_output_reg = 0x0003	left_alu_reg = 0x0001 right_alu_reg = 0x0002 alu_output_reg = 0x0003
Alu_normal_add	0x8001	Addition (left_alu_reg+ right_alu_reg)	left_alu_reg = 0x0001 1. right_alu_reg = 0x0002 2. right_alu_reg = 0xFFFF 3. right_alu_reg = 0x7FFF 4. right_alu_reg = 0x0002, Version Register = 0x8210	1. alu_output_reg = 0x0003 2. alu_output_reg = 0x0000, carry ignored 3. Status Register = 0x0102, Error state 4. alu_output_reg = 0x0003
Alu_normal_sub	0x8002	Subtraction	right_alu_reg = 0x0001	

		(left_alu_reg-right_alu_reg)	1. left_alu_reg = 0x0003 2. left_alu_reg = 0x8000 3. left_alu_reg = 0x0003, Version Register = 0x8210	1. alu_output_reg = 0x0002 2. Status Register = 0x0102, Error state 3. alu_output_reg = 0x0002
Alu_normal_move_left	0x8003	Move to left (left_alu_reg= alu_output_reg)	1. alu_output_reg = 0x0001 2. alu_output_reg = 0x0001, Version Register = 0x8210	1. left_alu_reg = 0x0001 2. Status Register = 0x0208, Export Violation
Alu_normal_move_right	0x8004	Move to Right (right_alu_reg= alu_output_reg)	1. alu_output_reg = 0x0001 2. alu_output_reg = 0x0001, Version Register = 0x8210	1. right_alu_reg = 0x0001 2. Status Register = 0x0208, Export Violation
Alu_normal_swap	0x8005	Swap (left_alu_reg= right_alu_reg)	1. left_alu_reg = 0x0003, right_alu_reg = 0x0002 2. left_alu_reg = 0x0003, right_alu_reg = 0x0002, Version Register = 0x8210	1. left_alu_reg = 0x0002 right_alu_reg = 0x0003 2. Status Register = 0x0208, Export Violation
Alu_normal_shift_left	0x8006	Shift left (left_alu_reg << right_alu_reg)	left_alu_reg = 0xFFFF 1. right_alu_reg = 0x0000 2. right_alu_reg = 0x0001 3. right_alu_reg = 0x000F 4. right_alu_reg = 0x0010 5. right_alu_reg = 0xFFFF 6. right_alu_reg = 0x0000, Version Register = 0x8210	1. alu_output_reg = 0xFFFF 2. alu_output_reg = 0xFFFE 3. alu_output_reg = 0x8000 4. alu_output_reg = 0x0000 5. alu_output_reg = 0x0000 6. Status Register = 0x0208, Export Violation

Alu_normal_shift_right	0x8007	Shift Right (left_alu_reg >> right_alu_reg)	left_alu_reg = 0xFFFF 1. right_alu_reg = 0x0000 2. right_alu_reg = 0x0001 3. right_alu_reg = 0x000F 4. right_alu_reg = 0x0010 5. right_alu_reg = 0xFFFF 6. right_alu_reg = 0x0000, Version Register = 0x8210	1. alu_output_reg = 0xFFFF 2. alu_output_reg = 0x7FFF 3. alu_output_reg = 0x0001 4. alu_output_reg = 0x0000 5. alu_output_reg = 0x0000 6. Status Register = 0x0208, Export Violation
Alu_normal_reserved	0x800F	Reserved	1. Version Register = 0x0210 2. Version Register = 0x8210	1. Status Register = 0x0102, Error state 2. Status Register = 0x0208, Export Violation

1.3. Test Table - Valid (Command Register) is 1'b0:

1.3.1 The signal export_disable is 1'bx

1.3.1 The alu_output_reg, left_alu_reg, right_alu_reg if not mentioned in test description is to be set to 0x0000, 0x0001, 0x0002 respectively.

Test name	Command Register	ALU operation	Test Description	Expected Result
Alu_normal_no_cmd	0x8000	No command	left_alu_reg = 0x0001 right_alu_reg = 0x0002 alu_output_reg = 0x0003	No command is performed, left_alu_reg = 0x0001 right_alu_reg = 0x0002 alu_output_reg = 0x0003
Alu_normal_add	0x8001	Addition (left_alu_reg+ right_alu_reg)	left_alu_reg = 0x0001 right_alu_reg = 0x0002	No command is performed, left_alu_reg = 0x0001 right_alu_reg = 0x0002 alu_output_reg = 0x0000

Alu_normal_sub	0x8002	Subtraction (left_alu_reg- right_alu_reg)	right_alu_reg = 0x0001 left_alu_reg = 0x0003	No command is performed, left_alu_reg = 0x0001 right_alu_reg = 0x0002 alu_output_reg = 0x0000
Alu_normal_move_left	0x8003	Move to left (left_alu_reg= alu_output_reg)	alu_output_reg = 0x0001	No command is performed, left_alu_reg = 0x0001 right_alu_reg = 0x0002 alu_output_reg = 0x0001
Alu_normal_move_right	0x8004	Move to Right (right_alu_reg= alu_output_reg)	alu_output_reg = 0x0001	No command is performed, left_alu_reg = 0x0001 right_alu_reg = 0x0002 alu_output_reg = 0x0001
Alu_normal_swap	0x8005	Swap (left_alu_reg= right_alu_reg)	left_alu_reg = 0x0003, right_alu_reg = 0x0002	No command is performed, left_alu_reg = 0x0003 right_alu_reg = 0x0002 alu_output_reg = 0x0000
Alu_normal_shift_left	0x8006	Shift left (left_alu_reg << right_alu_reg)	left_alu_reg = 0x0003 right_alu_reg = 0x0000	No command is performed, left_alu_reg = 0x0003 right_alu_reg = 0x0000 alu_output_reg = 0x0000
Alu_normal_shift_right	0x8007	Shift Right (left_alu_reg >> right_alu_reg)	left_alu_reg = 0x0003 right_alu_reg = 0x0000	No command is performed, left_alu_reg = 0x0003 right_alu_reg = 0x0000 alu_output_reg = 0x0000
Alu_normal_reserved	0x800F	Reserved	1. Version Register = 0x0210 2. Version Register = 0x8210	No command is performed, left_alu_reg = 0x0003 right_alu_reg = 0x0000 alu_output_reg = 0x0000

2. Verichip in **Reset** State (Section 6.1, 5.1, 5.2, 5.3, 5.5, 5.6, 5.7)

2.1. Objective:

2.1.1 Verify that No commands are Executed in the Reset state.

2.1.2 Registers values should be unchanged

2.1.3 The alu_output_reg, left_alu_reg and right_alu_reg are to be set to 0x0000, 0x0001, 0x0002 respectively, to check if they hold values.

2.1.4 All ALU operations should be performed and alu_output_reg should be Unchanged.

2.1.5 State machine remains in Reset state (Operation Status = 4'h0) after every test.

2.1.6 State machine must not change states to Error.

Valid	export_disable	Description
1’b0	1’b0	No commands are performed; registers should hold their respective values: alu_output_reg = 0x0000 left_alu_reg = 0x0001 right_alu_reg = 0x0002
1’b0	1’b1	No commands are performed; registers should hold their respective values: alu_output_reg = 0x0000 left_alu_reg = 0x0001 right_alu_reg = 0x0002
1’b1	1’b0	No commands are performed; registers should hold their respective values: alu_output_reg = 0x0000 left_alu_reg = 0x0001 right_alu_reg = 0x0002
1’b1	1’b1	No commands are performed; registers should hold their respective values: alu_output_reg = 0x0000 left_alu_reg = 0x0001 right_alu_reg = 0x0002

End of the Document

Thank You