Team Fnatic - ALU Test Plan

Test:	**ALU in Reset State**
Section:	5.0, 5.2, 5.3, 5.5, 5.6, 5.7, 6.1
What to test:	1. Ensure that all ALU_Left, ALU_Right and ALU_Out have different values and attempt to do all commands with valid bit = 0. No commands should be executed and verify all ALU_Left, ALU_Right and ALU_Out have the same values as before, and the state machine had remained in the reset state. 2. Ensure that all ALU_Left, ALU_Right and ALU_Out have different values and attempt to do all commands with valid bit = 1. No commands should be executed because commands are ignored in the error state and ALU_Left, ALU_Right and ALU_Out should have the same values as before, and the state machine has remained in the reset state. 3. Ensure that all ALU_Left, ALU_Right and ALU_Out have different values and attempt to do all commands with valid bit = 0 and export_dissable asserted. No commands should be executed and verify all ALU_Left, ALU_Right and ALU_Out have the same values as before, and the state machine had remained in the reset state. 4. Ensure that all ALU_Left, ALU_Right and ALU_Out have different values and attempt to do all commands with valid bit = 1 and export_disable asserted. No commands should be executed because commands are ignored in the error state and ALU_Left, ALU_Right and ALU_Out should have the same values as before, and the state machine had remained in the reset state.

Test:	**ALU in Normal State**
Section:	5.0, 5.2, 5.3, 5.5, 5.6, 5.7, 6.2
What to test:	**When the Valid bit is 0:** 1. Start by writing different values into ALU_Left, ALU_Right and ALU_Out registers. When the valid bit is 0 and export_disable asserted, no command should be executed, the registers should read back the initial values and the state machine remains in the normal state. This applies to all the operations by the Command Register.

2. Start by writing different values into ALU_Left, ALU_Right and ALU_Out registers. When the valid bit is 0 and export_disable not asserted, no command should be executed the registers should read back the initial values and the state machine remains in the normal state. This applies to all the valid operations by the Command Register.

When the Valid bit is 1:

1. **Addition:** (Test the following with export_disable asserted and not asserted)

a. The ALU should perform addition only when the command register is written with the value 16'h8001, and the valid bit is 1.

b. Start by writing different values into ALU_Left and ALU_Right registers and perform the addition and verify that ALU_Out reads the correct sum.

c. Check for corner cases by giving appropriate values to ALU_Left and ALU_Right, to check if overflow occurs. In case overflow occurs, the state machine should transition to Error state, or else it stays in the Normal State.

d. Since Verichip supports signed arithmetic, check the sum of 2 numbers with different signs and also check if overflow occurs for certain inputs. In case overflow occurs, the state machine should transition to Error state or else it stays in the Normal State.

2. Subtraction: (Test the following with export_disable asserted and not asserted)

a. The ALU should perform subtraction only when the command register is written with the value 16'h8002 and the valid bit is 1.

b. Start by writing different values into ALU_Left and ALU_Right registers and perform the subtraction and verify that ALU_Out reads the correct difference.

c. Check for corner cases by giving appropriate values to ALU_Left and ALU_Right, to check if overflow occurs. In case overflow occurs, the state machine should transition to Error state or else it stays in the Normal State.

d. Since Verichip supports signed arithmetic, check the difference of 2 numbers with different signs and also check if overflow occurs for certain inputs. In case overflow occurs, the state machine should transition to Error state, or else it stays in the Normal State.

	3. Move to Left: a. The ALU should perform Move to Left only when the command register is written with the value 16'h8003 and the valid bit is 1. b. Test with export_disable asserted and not asserted. c. When export_disable is asserted, the state machine should transition to the export violation state. d. When export_disable is not asserted, start by writing different values into ALU_Left and ALU_Right registers and perform any operation to generate some value at ALU_Out. Now perform the move to left operation and then ALU_Left register should read the value generated at ALU_Out and not the initial value (unless both are the same). The state machine should stay in the normal state. **4. Move to Right:** a. The ALU should perform Move to Left only when the command register is written with the value 16'h8004 and the valid bit is 1. b. Test with export_disable asserted and not asserted. c. When export_disable is asserted, the state machine should transition to the export violation state. d. When export_disable is not asserted, start by writing different values into ALU_Left and ALU_Right registers and perform any operation to generate some value at ALU_Out. Now perform the move to right operation and then ALU_Right register should read the value generated at ALU_Out and not the initial value (unless both are the same). The state machine should stay in the normal state. **5. Swap:** a. The ALU should perform Move to Left only when the command register is written with the value 16'h8005, and the valid bit is 1 b. Test with export_disable asserted and not asserted. c. When export_disable is asserted, the state machine should transition to the export violation state. d. When export_disable is not asserted, start by writing different values into ALU_Left and ALU_Right registers. Now perform the swap operation and then ALU_Right register should read the initial value of ALU_Left register and the ALU_Left register should read the initial value of ALU_Right register. The state machine should stay in the normal state.

	6. Shift Left: a. The ALU should perform Move to Left only when the command register is written with the value 16'h8006 and the valid bit is 1 b. Test with export_disable asserted and not asserted. c. When export_disable is asserted, the state machine should transition to the export violation state. d. When export_disable is not asserted, start by writing different values into ALU_Left and ALU_Right registers. Now perform the Shift Left operation and then ALU_Out register should read ALU_Left shifted left by ALU_Right times (for eg, if ALU_Left = 16'hAAAA and ALU_Right = 16'h0001, then after left shift operation, the ALU_Out should read 16'h5554). The state machine should stay in the normal state. **7. Shift Right:** a. The ALU should perform Move to Left only when the command register is written with the value 16'h8007 and the valid bit is 1 b. Test with export_disable asserted and not asserted. c. When export_disable is asserted, the state machine should transition to the export violation state. d. When export_disable is not asserted, start by writing different values into ALU_Left and ALU_Right registers. Now perform the Shift Right operation and then ALU_Out register should read ALU_Left shifted right by ALU_Right times (for eg, if ALU_Left = 16'hBBBB and ALU_Right = 16'h0002, then after right shift operation, the ALU_Out should read 16'h2EEE). The state machine should stay in the normal state. **8**. Start by writing different values into ALU_Left and ALU_Right registers. The ALU should not execute any command when the command register is written 16'h8000. The state machine should stay in normal state if export_disable is not asserted, if asserted, it should transition to export violation state. The ALU Left, Right, Out should read the same values before attempting to run the command. **9**. Start by writing different values into ALU_Left and ALU_Right registers. In case of a reserved command being executed (for example, if the command register is written a value greater than 16'h8007) it is considered as an invalid command and on encountering such commands, the state machine should transition to Error state if export_disable is not asserted. In this case read back and make sure the ALU Left, Right and Out read the same values as before. If export_disable is asserted on encountering invalid command, the state machine should transition to Export Violation state.