# LAB # 03

Name: Saqib Khan

UT EID: sak2454

**APPROACH TO FINDING BUGS:**

The following constraints were added:

1) Opcode must be with 0-31 (5 bits)
2) For (signed subtraction) sub, CIN must be equal to 1
3) For (unsigned subtraction) subu, CIN must be equal to 1

The approach to finding bugs in the ALU is as follows:

1) Generate expected results locally in the scoreboard.sv file using the same inputs (A, B, CIN, reset) and generating equivalent output to ALU (COUT, VOUT, OUT).
2) The expected results are compared to the actual results.
3) Three different error logs are generated based on difference in COUT, VOUT and OUT.
4) Different error logs can help in narrowing down the root cause of the problem in netlist.

**LIST OF BUGS:**

1) VOUT is not equal to 0 when RST == 1 for the following opcodes
00001, 00111,00011, 00101 {Logic Operations}
01100, 01000, 01101, 01110,01011,01111,01010 {Compare Operations}
10010,10011,10101,10001,10000,10111,10110 {Arithmetic Operations}
11010,11100,11101,11000,11001,11111,11110 {Shift Operations}

## A) Logic Operations:

**i) A OR B (00111)**

**ii) A XOR B (00011)**

**iii) NOT A (00000)**

The result of “Not A” is incorrect.

```
# Actual Input Values
#  RST =0 A =10000011001011101001010101110011 B
=01011001011000001011000000000001 CIN =0 Opcode =00000
# Actual Outputs
#  COUT =0 VOUT =0 OUT =01111100110100000110101010001100
# Expected Output
#  COUT =0 VOUT =0 OUT =01111100110100010110101010001100
```

**iv) A AND B (00101)**

Sign bit of the resulting A AND B is incorrect

```
# Actual Input Values
#  RST =0 A =11010100110001101110111010101001 B
=11011001010110110001001111001000 CIN =1 Opcode =00101
# Actual Outputs
#  COUT =0 VOUT =0 OUT =01010000010000100000001010001000
# Expected Output
#  COUT =0 VOUT =0 OUT =11010000010000100000001010001000
```

**v) Undefined Opcodes:**

a) Undefined Opcode should set COUT=VOUT=OUT=0

```
# Actual Input Values
#  RST =0 A =01101001011100010110100011010100 B
=11001101010110011110010100001111 CIN =1 Opcode =00010
# Actual Outputs
#  COUT =1 VOUT =1 OUT =11101101011110011110110111011111
# Expected Output
#  COUT =0 VOUT =0 OUT =00000000000000000000000000000000
```

B) **Compare Operations:**

i) **sle (A <= B) (01100)**
ii) **slt (A < B) (01001)**
iii) **sge (A >= B) (01110)**
iv) **sgt (A > B) (01011)**
v) **seq (A == B) (01111)**
vi) **sne (A != B) (01010)**
vii) **Undefined Opcodes**

C) **Arithmetic Operations:**

i) **Add (signed A + B) (10101)**
ii) **Addu (unsigned A + B) (10001)**
iii) **Sub (signed A – B) (10100)**
iv) **Subu (unsigned A – B) (10000)**

subu: The COUT of a unsigned subtraction must always be 1

```
# Actual Input Values
#  RST =0 A =10001011111100101100011100011001 B
=00000010011011100011111000011101 CIN =1 Opcode =10000
# Actual Outputs
#  COUT =0 VOUT =0 OUT =10001001100001001000100011111100
# Expected Output
#  COUT =1 VOUT =0 OUT =10001001100001001000100011111100
```

v) **Inc (signed A + 1) (10111)**
vi) **Dec (signed A – 1) (10110)**
vii) **Undefined Opcodes**

**D) Shift Operations:**

**i) sll (logic left shift) (11010)**

**ii) srl (logic right shift) (11011)**

The logic right shift is done by incorrect number of bits

```
# Actual Input Values
#  RST =0 A =11101010011111100100000010011110 B
=11010100011110010010011100011000 CIN =0 Opcode =11011
# Actual Outputs
#  COUT =0 VOUT =0 OUT =00000000111010100111111001000000
# Expected Output
#  COUT =0 VOUT =0 OUT =00000000000000000000000011101010
```

**iii) sla (arithmetic left shift) (11100)**

**iv) sra (arithmetic right shift) (11101)**

For arithmetic right shift COUT must always be 0.

```
# Actual Input Values
#  RST =0 A =11110000110101100010001011110000 B
=00000010110110010111111000010001 CIN =1 Opcode =11101
# Actual Outputs
#  COUT =1 VOUT =1 OUT =11111111111111111111100001101011
# Expected Output
#  COUT =0 VOUT =1 OUT =11111111111111111111100001101011
```

**v) slr (rotate left shift) (11000)**

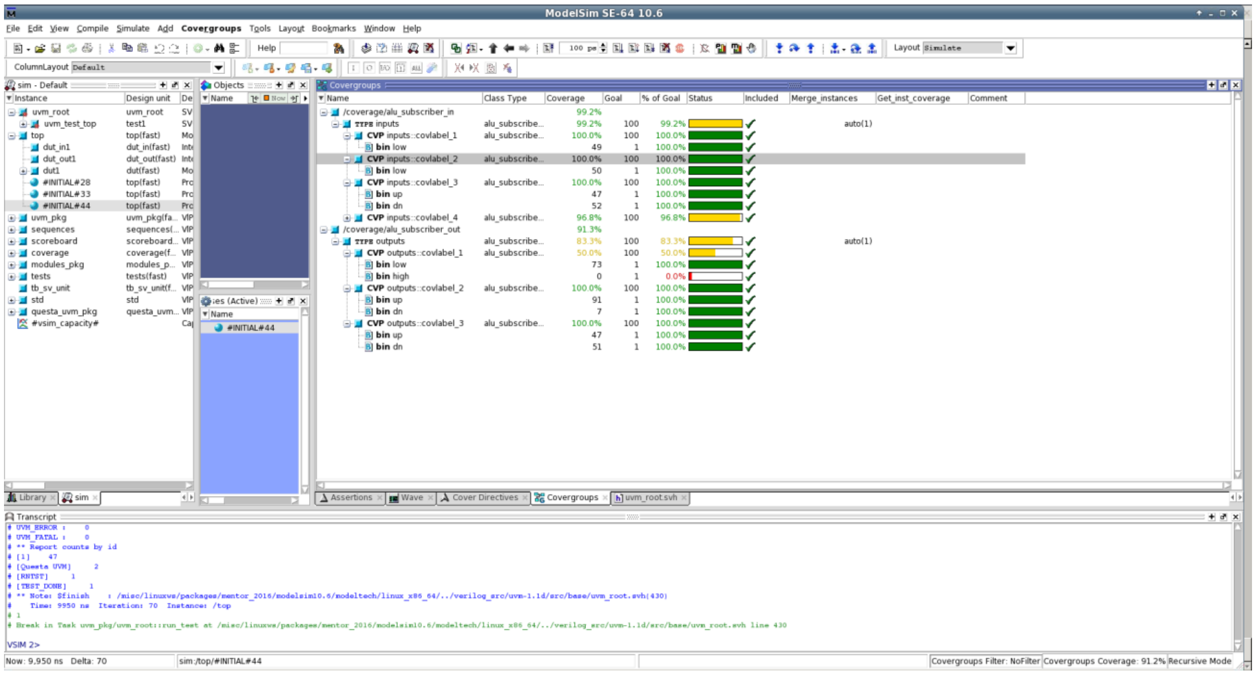
**vi) srr (rotate right shift) (11001)**

**vii) Undefined Opcodes**

The coverage was measured for all inputs and outputs:

Input Coverage:
CIN: 0 and 1
A: 0->2147483647
B: 0->2147483647
Opcode: 0->31

Output Coverage:
COUT: 0 and 1
VOUT: 0 and 1
OUT: low = {[0:2000]};
        high = {[2001:10000]};



| Instance | Design unit | De |
|---|---|---|
| uvm_root | uvm_root | SV |
| uvm_test_top | test1 | SV |
| top | top(fast) | Mo |
| dut_in1 | dut_in(fast) | Int |
| dut_out1 | dut_out(fast) | Int |
| dut1 | dut(fast) | Mo |
| #INITIAL#28 | top(fast) | Pro |
| #INITIAL#33 | top(fast) | Pro |
| #INITIAL#44 | top(fast) | Pro |
| uvm_pkg | uvm_pkg(fa... | VIP |
| sequences | sequences(... | VIP |
| scoreboard | scoreboard... | VIP |
| coverage | coverage(f... | VIP |
| modules_pkg | modules_p... | VIP |
| tests | tests(fast) | VIP |
| tb_sv_unit | tb_sv_unit(f... | VIP |
| std | std | VIP |
| questa_uvm_pkg | questa_uvm... | VIP |
| #vsim_capacity# | | Ca |

| Name | Class Type | Coverage | Goal | % of Goal | Status | Included | Merge_instances | Get_inst_coverage | Comment |
|---|---|---|---|---|---|---|---|---|---|
| /coverage/alu_subscriber_in | | 99.2% | | | | | | | |
| TYPE inputs | alu_subscribe... | 99.2% | 100 | 99.2% | | ✓ | | auto(1) | |
| CVP inputs::covlabel_1 | alu_subscribe... | 100.0% | 100 | 100.0% | | ✓ | | | |
| bin low | | 49 | 1 | 100.0% | | ✓ | | | |
| CVP inputs::covlabel_2 | alu_subscribe... | 100.0% | 100 | 100.0% | | ✓ | | | |
| bin low | | 50 | 1 | 100.0% | | ✓ | | | |
| CVP inputs::covlabel_3 | alu_subscribe... | 100.0% | 100 | 100.0% | | ✓ | | | |
| bin up | | 47 | 1 | 100.0% | | ✓ | | | |
| bin dn | | 52 | 1 | 100.0% | | ✓ | | | |
| CVP inputs::covlabel_4 | alu_subscribe... | 96.8% | 100 | 96.8% | | ✓ | | | |
| /coverage/alu_subscriber_out | | 91.3% | | | | | | | |
| TYPE outputs | alu_subscribe... | 83.3% | 100 | 83.3% | | ✓ | | auto(1) | |
| CVP outputs::covlabel_1 | alu_subscribe... | 50.0% | 100 | 50.0% | | ✓ | | | |
| bin low | | 73 | 1 | 100.0% | | ✓ | | | |
| bin high | | 0 | 1 | 0.0% | | ✓ | | | |
| CVP outputs::covlabel_2 | alu_subscribe... | 100.0% | 100 | 100.0% | | ✓ | | | |
| bin up | | 91 | 1 | 100.0% | | ✓ | | | |
| bin dn | | 7 | 1 | 100.0% | | ✓ | | | |
| CVP outputs::covlabel_3 | alu_subscribe... | 100.0% | 100 | 100.0% | | ✓ | | | |
| bin up | | 47 | 1 | 100.0% | | ✓ | | | |
| bin dn | | 51 | 1 | 100.0% | | ✓ | | | |

```
# UVM_ERROR :    0
# UVM_FATAL :    0
# ** Report counts by id
# [1]    47
# [Questa UVM]     2
# [RNTST]     1
# [TEST_DONE]     1
# ** Note: $finish    : /misc/linuxws/packages/mentor_2016/modelsim10.6/modeltech/linux_x86_64/../verilog_src/uvm-1.1d/src/base/uvm_root.svh(430)
#    Time: 9950 ns  Iteration: 70  Instance: /top
# 1
# Break in Task uvm_pkg/uvm_root::run_test at /misc/linuxws/packages/mentor_2016/modelsim10.6/modeltech/linux_x86_64/../verilog_src/uvm-1.1d/src/base/uvm_root.svh line 430

VSIM 2>
```

Now: 9,950 ns   Delta: 70   sim:/top/#INITIAL#44   Covergroups Filter: NoFilter   Covergroups Coverage: 91.2%   Recursive Mode

*Figure-1: Cover Report for ALU test bench*