**Arithmetic Test Cases:**

**ADD**

* Test where both are zero
* Test to see if the operation is signed
* Test where a register is the negative of the other to see if 0 is obtained in rd
* Test where one register is 'b0

**SUB**

* Test that the instruction correctly handles the edge case of subtracting 0
* Test that the instruction performs the correct subtraction operation between two registers
* Test the negative number and positive number combinations
* Test that the instruction correctly updates the rd
* Test that the instruction correctly handles overflow and underflow

**SLL**

* Test with random values and shift amounts within the range of 0 to 31
* Input value is a positive number, and the shift amount is 1
* Input value is zero and the shift amount is zero
* Test with some input value and shift amount equal to 31
* Input value is all ones, and the shift amount is zero
* Tests to check whether the lower 5 bits are getting extracted from the rs2

**SLT**

* Tested with basic values.
* Following are the updated test cases with different values (results unchanged):
  + rs1 = +45 and rs2 = -10. Sets rd = 0
  + rs1 = -50 and rs2 = -20. Sets rd = 1
  + rs1 = -60 and rs2 = +25. Sets rd = 1
  + rs1 = +30 and rs2 = +15. Sets rd = 0
  + rs1 = -5 and rs2 = 0. Sets rd = 1
  + rs1 = 0 and rs2 = 0. Sets rd = 0

**32. SLTU**

* Test for one operand zero
* Test for both operands equal
* Test for both operands negative (unsigned, so converted to positive)
* Test for both operands 0

**33. XOR**

* Test that the instruction correctly handles the edge case of XOR-ing with 1
* Test that the instruction correctly handles the edge case of XOR-ing with 0
* Test that the instruction correctly updates the rd
* Test that the instruction performs the correct bitwise XOR operation between two registers

**34. SRL**

* Test with random values and shift amounts within the range of 0 to 31
* Input value is a positive number, and the shift amount is 1
* Test with some input value and shift amount equal to 31
* Input value is all ones, and the shift amount is zero
* Input value is zero, and the shift amount is zero
* Tests to check whether the lower 5 bits are getting extracted from the rs2

**SRA**

* Tested with basic shifting
* Tested with MSB=1 value
* Tested for the following updated values (results unchanged):
  + rs1 = -64 and rs2 = 3
  + rs1 = +56 and rs2 = 2
  + rs1 = -48 and rs2 = 5. Lower 5 bits are taken and treated as a positive decimal shift amount
  + rs1 = +40 and rs2 = 6. Lower 5 bits are taken and treated as a positive decimal shift amount
  + rs1 = +32 and rs2 = 1. Lower 5 bits are taken and treated as a positive decimal shift amount

**OR**

* Test for one register is ‘b0
* Test for one register is ‘b1
* One register ‘ba and other is ‘b4

**AND**

* One register all zero and other is all 1s
* One register all 1s and other is all zero