# Testbenching Report for logic_shifter

# Table of Contents

## Testbench Summary

| Component | Total Tests | Passed | Failed |
|---|---|---|---|
| logic_shifter_N2 | 16 | 16 | 0 |
| logic_shifter_N3 | 64 | 64 | 0 |
| logic_shifter_N4 | 128 | 128 | 0 |
| logic_shifter_N5 | 512 | 512 | 0 |
| logic_shifter_N6 | 1024 | 1024 | 0 |
| logic_shifter_N7 | 2048 | 2048 | 0 |
| logic_shifter_N8 | 2599 | 2599 | 0 |

## Testbench for logic_shifter with parameter(s) N2

Total tests: 16

Passed tests: 16

Failed tests: 0

| Test Case | Input data_in | Input shift_amount | Input direction | Output data_out (Actual) | Expected data_out | Status |
|---|---|---|---|---|---|---|
| 9 | 00 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 00 (bin) / 0 (dec) | 0 (dec) | Passed |
| 0 | 00 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 00 (bin) / 0 (dec) | 0 (dec) | Passed |
| 6 | 10 (bin) / 2 (dec) | 0 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 10 (bin) / 2 (dec) | 2 (dec) | Passed |
| 8 | 01 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 0 (bin) / 0 (dec) | 10 (bin) / 2 (dec) | 2 (dec) | Passed |
| 11 | 01 (bin) / 1 (dec) | 0 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 01 (bin) / 1 (dec) | 1 (dec) | Passed |
| 5 | 11 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 11 (bin) / 3 (dec) | 3 (dec) | Passed |
| 7 | 10 (bin) / 2 (dec) | 0 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 10 (bin) / 2 (dec) | 2 (dec) | Passed |
| 2 | 11 (bin) / 3 (dec) | 1 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 01 (bin) / 1 (dec) | 1 (dec) | Passed |
| 15 | 01 (bin) / 1 (dec) | 0 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 01 (bin) / 1 (dec) | 1 (dec) | Passed |
| 1 | 01 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 00 (bin) / 0 (dec) | 0 (dec) | Passed |
| 3 | 10 (bin) / 2 (dec) | 1 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 01 (bin) / 1 (dec) | 1 (dec) | Passed |
| 13 | 00 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 0 (bin) / 0 (dec) | 00 (bin) / 0 (dec) | 0 (dec) | Passed |
| 4 | 10 (bin) / 2 (dec) | 1 (bin) / 1 (dec) | 0 (bin) / 0 (dec) | 00 (bin) / 0 (dec) | 0 (dec) | Passed |
| 14 | 11 (bin) / 3 (dec) | 1 (bin) / 1 (dec) | 0 (bin) / 0 (dec) | 10 (bin) / 2 (dec) | 2 (dec) | Passed |
| 12 | 11 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 11 (bin) / 3 (dec) | 3 (dec) | Passed |
| 10 | 00 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 00 (bin) / 0 (dec) | 0 (dec) | Passed |

## Rule: LogicalShifterRule

Input Variables: data_in, shift_amount, direction

Output Variables: data_out

Bit Width: 4

Pattern: StringMatchPattern

```
            def matches(self, filename):
                #print(self.pattern, filename)
                return self.pattern == filename
```

Generate expected values function:

```
            def generate_expected(self, test_case):
                max_val = (1 << self.bit_width) - 1
                if test_case["direction"] == 0:
                    result = (test_case["data_in"] << test_case["shift_amount"]) & max_val
                else:  # right shift
                    result = (test_case["data_in"] >> test_case["shift_amount"]) & max_val

                outs = {
                    "data_out": result
                }
                return outs
```

## Testbench for logic_shifter with parameter(s) N3

Total tests: 64

Passed tests: 64

Failed tests: 0

| Test Case | Input data_in | Input shift_amount | Input direction | Output data_out (Actual) | Expected data_out | Status |
|---|---|---|---|---|---|---|
| 8 | 001 (bin) / 1 (dec) | 11 (bin) / 3 (dec) | 1 (bin) / 1 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 17 | 100 (bin) / 4 (dec) | 00 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 100 (bin) / 4 (dec) | 4 (dec) | Passed |
| 55 | 100 (bin) / 4 (dec) | 00 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 100 (bin) / 4 (dec) | 4 (dec) | Passed |
| 52 | 010 (bin) / 2 (dec) | 11 (bin) / 3 (dec) | 1 (bin) / 1 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 60 | 111 (bin) / 7 (dec) | 00 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 111 (bin) / 7 (dec) | 7 (dec) | Passed |
| 0 | 110 (bin) / 6 (dec) | 00 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 110 (bin) / 6 (dec) | 6 (dec) | Passed |
| 36 | 101 (bin) / 5 (dec) | 11 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 31 | 000 (bin) / 0 (dec) | 11 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 6 | 011 (bin) / 3 (dec) | 00 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 011 (bin) / 3 (dec) | 3 (dec) | Passed |
| 49 | 110 (bin) / 6 (dec) | 10 (bin) / 2 (dec) | 0 (bin) / 0 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 24 | 101 (bin) / 5 (dec) | 10 (bin) / 2 (dec) | 0 (bin) / 0 (dec) | 100 (bin) / 4 (dec) | 4 (dec) | Passed |
| 23 | 011 (bin) / 3 (dec) | 10 (bin) / 2 (dec) | 1 (bin) / 1 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 14 | 010 (bin) / 2 (dec) | 10 (bin) / 2 (dec) | 1 (bin) / 1 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 2 | 010 (bin) / 2 (dec) | 00 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 010 (bin) / 2 (dec) | 2 (dec) | Passed |
| 58 | 100 (bin) / 4 (dec) | 11 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 28 | 000 (bin) / 0 (dec) | 00 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 48 | 100 (bin) / 4 (dec) | 11 (bin) / 3 (dec) | 1 (bin) / 1 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 26 | 001 (bin) / 1 (dec) | 11 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 30 | 110 (bin) / 6 (dec) | 11 (bin) / 3 (dec) | 1 (bin) / 1 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 54 | 111 (bin) / 7 (dec) | 11 (bin) / 3 (dec) | 1 (bin) / 1 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 40 | 000 (bin) / 0 (dec) | 10 (bin) / 2 (dec) | 1 (bin) / 1 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 9 | 110 (bin) / 6 (dec) | 10 (bin) / 2 (dec) | 1 (bin) / 1 (dec) | 001 (bin) / 1 (dec) | 1 (dec) | Passed |
| 35 | 010 (bin) / 2 (dec) | 00 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 010 (bin) / 2 (dec) | 2 (dec) | Passed |

| Test Case | Input data_in | Input shift_amount | Input direction | Output data_out (Actual) | Expected data_out | Status |
|---|---|---|---|---|---|---|
| 51 | 011 (bin) / 3 (dec) | 01 (bin) / 1 (dec) | 0 (bin) / 0 (dec) | 110 (bin) / 6 (dec) | 6 (dec) | Passed |
| 56 | 000 (bin) / 0 (dec) | 00 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 000 (bin) / 0 (dec) | 0 (dec) | Passed |

## Rule: LogicalShifterRule

Input Variables: data_in, shift_amount, direction

Output Variables: data_out

Bit Width: 4

Pattern: StringMatchPattern

```
            def matches(self, filename):
                #print(self.pattern, filename)
                return self.pattern == filename
```

Generate expected values function:

```
            def generate_expected(self, test_case):
                max_val = (1 << self.bit_width) - 1
                if test_case["direction"] == 0:
                    result = (test_case["data_in"] << test_case["shift_amount"]) & max_val
                else:  # right shift
                    result = (test_case["data_in"] >> test_case["shift_amount"]) & max_val

                outs = {
                    "data_out": result
                }
                return outs
```

## Testbench for logic_shifter with parameter(s) N4

Total tests: 128

Passed tests: 128

Failed tests: 0

| Test Case | Input data_in | Input shift_amount | Input direction | Output data_out (Actual) | Expected data_out | Status |
|---|---|---|---|---|---|---|
| 31 | 1100 (bin) / 12 (dec) | 10 (bin) / 2 (dec) | 0 (bin) / 0 (dec) | 0000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 5 | 1110 (bin) / 14 (dec) | 11 (bin) / 3 (dec) | 1 (bin) / 1 (dec) | 0001 (bin) / 1 (dec) | 1 (dec) | Passed |
| 77 | 1000 (bin) / 8 (dec) | 00 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 1000 (bin) / 8 (dec) | 8 (dec) | Passed |
| 64 | 0101 (bin) / 5 (dec) | 01 (bin) / 1 (dec) | 0 (bin) / 0 (dec) | 1010 (bin) / 10 (dec) | 10 (dec) | Passed |
| 41 | 0100 (bin) / 4 (dec) | 01 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 0010 (bin) / 2 (dec) | 2 (dec) | Passed |
| 27 | 0010 (bin) / 2 (dec) | 01 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 0001 (bin) / 1 (dec) | 1 (dec) | Passed |
| 93 | 0100 (bin) / 4 (dec) | 00 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 0100 (bin) / 4 (dec) | 4 (dec) | Passed |
| 3 | 1010 (bin) / 10 (dec) | 10 (bin) / 2 (dec) | 1 (bin) / 1 (dec) | 0010 (bin) / 2 (dec) | 2 (dec) | Passed |
| 111 | 1100 (bin) / 12 (dec) | 11 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 0000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 82 | 1010 (bin) / 10 (dec) | 10 (bin) / 2 (dec) | 0 (bin) / 0 (dec) | 1000 (bin) / 8 (dec) | 8 (dec) | Passed |
| 36 | 0010 (bin) / 2 (dec) | 11 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 0000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 37 | 0100 (bin) / 4 (dec) | 00 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 0100 (bin) / 4 (dec) | 4 (dec) | Passed |
| 117 | 0011 (bin) / 3 (dec) | 10 (bin) / 2 (dec) | 1 (bin) / 1 (dec) | 0000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 68 | 0110 (bin) / 6 (dec) | 01 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 0011 (bin) / 3 (dec) | 3 (dec) | Passed |
| 15 | 0011 (bin) / 3 (dec) | 00 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 0011 (bin) / 3 (dec) | 3 (dec) | Passed |
| 54 | 1101 (bin) / 13 (dec) | 01 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 0110 (bin) / 6 (dec) | 6 (dec) | Passed |
| 72 | 1101 (bin) / 13 (dec) | 00 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 1101 (bin) / 13 (dec) | 13 (dec) | Passed |
| 14 | 0001 (bin) / 1 (dec) | 01 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 0000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 51 | 0110 (bin) / 6 (dec) | 11 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 0000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 122 | 1011 (bin) / 11 (dec) | 10 (bin) / 2 (dec) | 1 (bin) / 1 (dec) | 0010 (bin) / 2 (dec) | 2 (dec) | Passed |
| 44 | 1101 (bin) / 13 (dec) | 10 (bin) / 2 (dec) | 0 (bin) / 0 (dec) | 0100 (bin) / 4 (dec) | 4 (dec) | Passed |
| 26 | 1101 (bin) / 13 (dec) | 00 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 1101 (bin) / 13 (dec) | 13 (dec) | Passed |
| 94 | 0111 (bin) / 7 (dec) | 11 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 1000 (bin) / 8 (dec) | 8 (dec) | Passed |

| Test Case | Input data_in | Input shift_amount | Input direction | Output data_out (Actual) | Expected data_out | Status |
|---|---|---|---|---|---|---|
| 32 | 1100 (bin) / 12 (dec) | 00 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 1100 (bin) / 12 (dec) | 12 (dec) | Passed |
| 11 | 0111 (bin) / 7 (dec) | 10 (bin) / 2 (dec) | 0 (bin) / 0 (dec) | 1100 (bin) / 12 (dec) | 12 (dec) | Passed |

## Rule: LogicalShifterRule

Input Variables: data_in, shift_amount, direction

Output Variables: data_out

Bit Width: 4

Pattern: StringMatchPattern

```
            def matches(self, filename):
                #print(self.pattern, filename)
                return self.pattern == filename
```

Generate expected values function:

```
            def generate_expected(self, test_case):
                max_val = (1 << self.bit_width) - 1
                if test_case["direction"] == 0:
                    result = (test_case["data_in"] << test_case["shift_amount"]) & max_val
                else:  # right shift
                    result = (test_case["data_in"] >> test_case["shift_amount"]) & max_val

                outs = {
                    "data_out": result
                }
                return outs
```

## Testbench for logic_shifter with parameter(s) N5

Total tests: 512

Passed tests: 512

Failed tests: 0

| Test Case | Input data_in | Input shift_amount | Input direction | Output data_out (Actual) | Expected data_out | Status |
|---|---|---|---|---|---|---|
| 187 | 01000 (bin) / 8 (dec) | 011 (bin) / 3 (dec) | 1 (bin) / 1 (dec) | 00001 (bin) / 1 (dec) | 1 (dec) | Passed |
| 19 | 10011 (bin) / 19 (dec) | 011 (bin) / 3 (dec) | 1 (bin) / 1 (dec) | 00010 (bin) / 2 (dec) | 2 (dec) | Passed |
| 183 | 11001 (bin) / 25 (dec) | 111 (bin) / 7 (dec) | 0 (bin) / 0 (dec) | 00000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 154 | 00111 (bin) / 7 (dec) | 110 (bin) / 6 (dec) | 0 (bin) / 0 (dec) | 00000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 171 | 10111 (bin) / 23 (dec) | 010 (bin) / 2 (dec) | 0 (bin) / 0 (dec) | 11100 (bin) / 28 (dec) | 28 (dec) | Passed |
| 470 | 00000 (bin) / 0 (dec) | 000 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 00000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 453 | 11010 (bin) / 26 (dec) | 100 (bin) / 4 (dec) | 1 (bin) / 1 (dec) | 00001 (bin) / 1 (dec) | 1 (dec) | Passed |
| 373 | 10000 (bin) / 16 (dec) | 111 (bin) / 7 (dec) | 0 (bin) / 0 (dec) | 00000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 424 | 00110 (bin) / 6 (dec) | 110 (bin) / 6 (dec) | 1 (bin) / 1 (dec) | 00000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 188 | 11101 (bin) / 29 (dec) | 110 (bin) / 6 (dec) | 1 (bin) / 1 (dec) | 00000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 180 | 11101 (bin) / 29 (dec) | 101 (bin) / 5 (dec) | 0 (bin) / 0 (dec) | 00000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 94 | 10111 (bin) / 23 (dec) | 000 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 10111 (bin) / 23 (dec) | 23 (dec) | Passed |
| 418 | 01011 (bin) / 11 (dec) | 010 (bin) / 2 (dec) | 0 (bin) / 0 (dec) | 01100 (bin) / 12 (dec) | 12 (dec) | Passed |
| 253 | 00001 (bin) / 1 (dec) | 000 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 00001 (bin) / 1 (dec) | 1 (dec) | Passed |
| 272 | 01111 (bin) / 15 (dec) | 011 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 11000 (bin) / 24 (dec) | 24 (dec) | Passed |
| 132 | 00111 (bin) / 7 (dec) | 101 (bin) / 5 (dec) | 0 (bin) / 0 (dec) | 00000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 485 | 01110 (bin) / 14 (dec) | 001 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 00111 (bin) / 7 (dec) | 7 (dec) | Passed |
| 459 | 11001 (bin) / 25 (dec) | 100 (bin) / 4 (dec) | 1 (bin) / 1 (dec) | 00001 (bin) / 1 (dec) | 1 (dec) | Passed |
| 241 | 00110 (bin) / 6 (dec) | 101 (bin) / 5 (dec) | 0 (bin) / 0 (dec) | 00000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 39 | 00001 (bin) / 1 (dec) | 100 (bin) / 4 (dec) | 1 (bin) / 1 (dec) | 00000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 44 | 00001 (bin) / 1 (dec) | 111 (bin) / 7 (dec) | 0 (bin) / 0 (dec) | 00000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 392 | 00001 (bin) / 1 (dec) | 000 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 00001 (bin) / 1 (dec) | 1 (dec) | Passed |
| 177 | 10101 (bin) / 21 (dec) | 000 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 10101 (bin) / 21 (dec) | 21 (dec) | Passed |

| Test Case | Input data_in | Input shift_amount | Input direction | Output data_out (Actual) | Expected data_out | Status |
|---|---|---|---|---|---|---|
| 378 | 01010 (bin) / 10 (dec) | 001 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 00101 (bin) / 5 (dec) | 5 (dec) | Passed |
| 360 | 01001 (bin) / 9 (dec) | 100 (bin) / 4 (dec) | 1 (bin) / 1 (dec) | 00000 (bin) / 0 (dec) | 0 (dec) | Passed |

## Rule: LogicalShifterRule

Input Variables: data_in, shift_amount, direction

Output Variables: data_out

Bit Width: 4

Pattern: StringMatchPattern

```
            def matches(self, filename):
                #print(self.pattern, filename)
                return self.pattern == filename
```

Generate expected values function:

```
            def generate_expected(self, test_case):
                max_val = (1 << self.bit_width) - 1
                if test_case["direction"] == 0:
                    result = (test_case["data_in"] << test_case["shift_amount"]) & max_val
                else:  # right shift
                    result = (test_case["data_in"] >> test_case["shift_amount"]) & max_val

                outs = {
                    "data_out": result
                }
                return outs
```

## Testbench for logic_shifter with parameter(s) N6

Total tests: 1024

Passed tests: 1024

Failed tests: 0

| Test Case | Input data_in | Input shift_amount | Input direction | Output data_out (Actual) | Expected data_out | Status |
|---|---|---|---|---|---|---|
| 93 | 100011 (bin) / 35 (dec) | 101 (bin) / 5 (dec) | 1 (bin) / 1 (dec) | 000001 (bin) / 1 (dec) | 1 (dec) | Passed |
| 622 | 110111 (bin) / 55 (dec) | 011 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 111000 (bin) / 56 (dec) | 56 (dec) | Passed |
| 96 | 011101 (bin) / 29 (dec) | 100 (bin) / 4 (dec) | 0 (bin) / 0 (dec) | 010000 (bin) / 16 (dec) | 16 (dec) | Passed |
| 7 | 100101 (bin) / 37 (dec) | 011 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 101000 (bin) / 40 (dec) | 40 (dec) | Passed |
| 74 | 000011 (bin) / 3 (dec) | 101 (bin) / 5 (dec) | 0 (bin) / 0 (dec) | 100000 (bin) / 32 (dec) | 32 (dec) | Passed |
| 371 | 101000 (bin) / 40 (dec) | 100 (bin) / 4 (dec) | 1 (bin) / 1 (dec) | 000010 (bin) / 2 (dec) | 2 (dec) | Passed |
| 280 | 001100 (bin) / 12 (dec) | 010 (bin) / 2 (dec) | 1 (bin) / 1 (dec) | 000011 (bin) / 3 (dec) | 3 (dec) | Passed |
| 608 | 110001 (bin) / 49 (dec) | 011 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 001000 (bin) / 8 (dec) | 8 (dec) | Passed |
| 982 | 010111 (bin) / 23 (dec) | 110 (bin) / 6 (dec) | 1 (bin) / 1 (dec) | 000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 707 | 111010 (bin) / 58 (dec) | 111 (bin) / 7 (dec) | 1 (bin) / 1 (dec) | 000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 316 | 010111 (bin) / 23 (dec) | 110 (bin) / 6 (dec) | 0 (bin) / 0 (dec) | 000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 773 | 011100 (bin) / 28 (dec) | 101 (bin) / 5 (dec) | 1 (bin) / 1 (dec) | 000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 779 | 111100 (bin) / 60 (dec) | 101 (bin) / 5 (dec) | 0 (bin) / 0 (dec) | 000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 97 | 111100 (bin) / 60 (dec) | 100 (bin) / 4 (dec) | 0 (bin) / 0 (dec) | 000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 408 | 111001 (bin) / 57 (dec) | 110 (bin) / 6 (dec) | 0 (bin) / 0 (dec) | 000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 185 | 110100 (bin) / 52 (dec) | 111 (bin) / 7 (dec) | 1 (bin) / 1 (dec) | 000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 221 | 110101 (bin) / 53 (dec) | 001 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 011010 (bin) / 26 (dec) | 26 (dec) | Passed |
| 249 | 100001 (bin) / 33 (dec) | 011 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 001000 (bin) / 8 (dec) | 8 (dec) | Passed |
| 419 | 101001 (bin) / 41 (dec) | 111 (bin) / 7 (dec) | 0 (bin) / 0 (dec) | 000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 884 | 011100 (bin) / 28 (dec) | 110 (bin) / 6 (dec) | 0 (bin) / 0 (dec) | 000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 464 | 011101 (bin) / 29 (dec) | 001 (bin) / 1 (dec) | 0 (bin) / 0 (dec) | 111010 (bin) / 58 (dec) | 58 (dec) | Passed |
| 37 | 010101 (bin) / 21 (dec) | 001 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 001010 (bin) / 10 (dec) | 10 (dec) | Passed |
| 934 | 101010 (bin) / 42 (dec) | 110 (bin) / 6 (dec) | 1 (bin) / 1 (dec) | 000000 (bin) / 0 (dec) | 0 (dec) | Passed |

| Test Case | Input data_in | Input shift_amount | Input direction | Output data_out (Actual) | Expected data_out | Status |
|---|---|---|---|---|---|---|
| 647 | 110011 (bin) / 51 (dec) | 010 (bin) / 2 (dec) | 1 (bin) / 1 (dec) | 001100 (bin) / 12 (dec) | 12 (dec) | Passed |
| 1 | 010100 (bin) / 20 (dec) | 101 (bin) / 5 (dec) | 1 (bin) / 1 (dec) | 000000 (bin) / 0 (dec) | 0 (dec) | Passed |

## Rule: LogicalShifterRule

Input Variables: data_in, shift_amount, direction

Output Variables: data_out

Bit Width: 4

Pattern: StringMatchPattern

```
            def matches(self, filename):
                #print(self.pattern, filename)
                return self.pattern == filename
```

Generate expected values function:

```
            def generate_expected(self, test_case):
                max_val = (1 << self.bit_width) - 1
                if test_case["direction"] == 0:
                    result = (test_case["data_in"] << test_case["shift_amount"]) & max_val
                else:  # right shift
                    result = (test_case["data_in"] >> test_case["shift_amount"]) & max_val

                outs = {
                    "data_out": result
                }
                return outs
```

## Testbench for logic_shifter with parameter(s) N7

Total tests: 2048

Passed tests: 2048

Failed tests: 0

| Test Case | Input data_in | Input shift_amount | Input direction | Output data_out (Actual) | Expected data_out | Status |
|---|---|---|---|---|---|---|
| 174 | 1010101 (bin) / 85 (dec) | 110 (bin) / 6 (dec) | 0 (bin) / 0 (dec) | 1000000 (bin) / 64 (dec) | 64 (dec) | Passed |
| 78 | 0011100 (bin) / 28 (dec) | 000 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 0011100 (bin) / 28 (dec) | 28 (dec) | Passed |
| 775 | 1000111 (bin) / 71 (dec) | 101 (bin) / 5 (dec) | 0 (bin) / 0 (dec) | 1100000 (bin) / 96 (dec) | 96 (dec) | Passed |
| 1079 | 1011010 (bin) / 90 (dec) | 111 (bin) / 7 (dec) | 1 (bin) / 1 (dec) | 0000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 1278 | 1100011 (bin) / 99 (dec) | 010 (bin) / 2 (dec) | 0 (bin) / 0 (dec) | 0001100 (bin) / 12 (dec) | 12 (dec) | Passed |
| 1093 | 1010001 (bin) / 81 (dec) | 001 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 0101000 (bin) / 40 (dec) | 40 (dec) | Passed |
| 1658 | 1110000 (bin) / 112 (dec) | 100 (bin) / 4 (dec) | 1 (bin) / 1 (dec) | 0000111 (bin) / 7 (dec) | 7 (dec) | Passed |
| 966 | 0100100 (bin) / 36 (dec) | 001 (bin) / 1 (dec) | 0 (bin) / 0 (dec) | 1001000 (bin) / 72 (dec) | 72 (dec) | Passed |
| 62 | 1001001 (bin) / 73 (dec) | 110 (bin) / 6 (dec) | 1 (bin) / 1 (dec) | 0000001 (bin) / 1 (dec) | 1 (dec) | Passed |
| 751 | 0000110 (bin) / 6 (dec) | 100 (bin) / 4 (dec) | 0 (bin) / 0 (dec) | 1100000 (bin) / 96 (dec) | 96 (dec) | Passed |
| 1122 | 1111110 (bin) / 126 (dec) | 110 (bin) / 6 (dec) | 0 (bin) / 0 (dec) | 0000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 905 | 1100110 (bin) / 102 (dec) | 111 (bin) / 7 (dec) | 1 (bin) / 1 (dec) | 0000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 1365 | 0010010 (bin) / 18 (dec) | 101 (bin) / 5 (dec) | 1 (bin) / 1 (dec) | 0000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 273 | 1011111 (bin) / 95 (dec) | 110 (bin) / 6 (dec) | 0 (bin) / 0 (dec) | 1000000 (bin) / 64 (dec) | 64 (dec) | Passed |
| 640 | 1000001 (bin) / 65 (dec) | 101 (bin) / 5 (dec) | 0 (bin) / 0 (dec) | 0100000 (bin) / 32 (dec) | 32 (dec) | Passed |
| 1034 | 1010000 (bin) / 80 (dec) | 101 (bin) / 5 (dec) | 0 (bin) / 0 (dec) | 0000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 39 | 1101010 (bin) / 106 (dec) | 110 (bin) / 6 (dec) | 0 (bin) / 0 (dec) | 0000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 1152 | 0001010 (bin) / 10 (dec) | 101 (bin) / 5 (dec) | 1 (bin) / 1 (dec) | 0000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 448 | 1100111 (bin) / 103 (dec) | 101 (bin) / 5 (dec) | 1 (bin) / 1 (dec) | 0000011 (bin) / 3 (dec) | 3 (dec) | Passed |
| 402 | 1000001 (bin) / 65 (dec) | 011 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 0001000 (bin) / 8 (dec) | 8 (dec) | Passed |
| 816 | 0011010 (bin) / 26 (dec) | 000 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 0011010 (bin) / 26 (dec) | 26 (dec) | Passed |
| 463 | 0001010 (bin) / 10 (dec) | 101 (bin) / 5 (dec) | 0 (bin) / 0 (dec) | 1000000 (bin) / 64 (dec) | 64 (dec) | Passed |
| 94 | 1001110 (bin) / 78 (dec) | 100 (bin) / 4 (dec) | 0 (bin) / 0 (dec) | 1100000 (bin) / 96 (dec) | 96 (dec) | Passed |

| Test Case | Input data_in | Input shift_amount | Input direction | Output data_out (Actual) | Expected data_out | Status |
|---|---|---|---|---|---|---|
| 207 | 0010000 (bin) / 16 (dec) | 000 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 0010000 (bin) / 16 (dec) | 16 (dec) | Passed |
| 1418 | 0001010 (bin) / 10 (dec) | 001 (bin) / 1 (dec) | 1 (bin) / 1 (dec) | 0000101 (bin) / 5 (dec) | 5 (dec) | Passed |

## Rule: LogicalShifterRule

Input Variables: data_in, shift_amount, direction

Output Variables: data_out

Bit Width: 4

Pattern: StringMatchPattern

```
            def matches(self, filename):
                #print(self.pattern, filename)
                return self.pattern == filename
```

Generate expected values function:

```
            def generate_expected(self, test_case):
                max_val = (1 << self.bit_width) - 1
                if test_case["direction"] == 0:
                    result = (test_case["data_in"] << test_case["shift_amount"]) & max_val
                else:  # right shift
                    result = (test_case["data_in"] >> test_case["shift_amount"]) & max_val

                outs = {
                    "data_out": result
                }
                return outs
```

## Testbench for logic_shifter with parameter(s) N8

Total tests: 2599

Passed tests: 2599

Failed tests: 0

| Test Case | Input data_in | Input shift_amount | Input direction | Output data_out (Actual) | Expected data_out | Status |
|---|---|---|---|---|---|---|
| 2203 | 11101011 (bin) / 235 (dec) | 111 (bin) / 7 (dec) | 1 (bin) / 1 (dec) | 00000001 (bin) / 1 (dec) | 1 (dec) | Passed |
| 1757 | 11110100 (bin) / 244 (dec) | 010 (bin) / 2 (dec) | 0 (bin) / 0 (dec) | 11010000 (bin) / 208 (dec) | 208 (dec) | Passed |
| 1695 | 01110101 (bin) / 117 (dec) | 010 (bin) / 2 (dec) | 1 (bin) / 1 (dec) | 00011101 (bin) / 29 (dec) | 29 (dec) | Passed |
| 1433 | 10101110 (bin) / 174 (dec) | 110 (bin) / 6 (dec) | 1 (bin) / 1 (dec) | 00000010 (bin) / 2 (dec) | 2 (dec) | Passed |
| 1873 | 11100011 (bin) / 227 (dec) | 100 (bin) / 4 (dec) | 1 (bin) / 1 (dec) | 00001110 (bin) / 14 (dec) | 14 (dec) | Passed |
| 2248 | 11100101 (bin) / 229 (dec) | 110 (bin) / 6 (dec) | 0 (bin) / 0 (dec) | 01000000 (bin) / 64 (dec) | 64 (dec) | Passed |
| 161 | 01010110 (bin) / 86 (dec) | 011 (bin) / 3 (dec) | 1 (bin) / 1 (dec) | 00001010 (bin) / 10 (dec) | 10 (dec) | Passed |
| 1489 | 01100011 (bin) / 99 (dec) | 010 (bin) / 2 (dec) | 1 (bin) / 1 (dec) | 00011000 (bin) / 24 (dec) | 24 (dec) | Passed |
| 1455 | 10011001 (bin) / 153 (dec) | 000 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 10011001 (bin) / 153 (dec) | 153 (dec) | Passed |
| 1663 | 00000010 (bin) / 2 (dec) | 011 (bin) / 3 (dec) | 1 (bin) / 1 (dec) | 00000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 2414 | 10111101 (bin) / 189 (dec) | 110 (bin) / 6 (dec) | 1 (bin) / 1 (dec) | 00000010 (bin) / 2 (dec) | 2 (dec) | Passed |
| 1772 | 11010111 (bin) / 215 (dec) | 010 (bin) / 2 (dec) | 0 (bin) / 0 (dec) | 01011100 (bin) / 92 (dec) | 92 (dec) | Passed |
| 177 | 01111101 (bin) / 125 (dec) | 111 (bin) / 7 (dec) | 1 (bin) / 1 (dec) | 00000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 1523 | 10000111 (bin) / 135 (dec) | 111 (bin) / 7 (dec) | 1 (bin) / 1 (dec) | 00000001 (bin) / 1 (dec) | 1 (dec) | Passed |
| 2468 | 01111111 (bin) / 127 (dec) | 000 (bin) / 0 (dec) | 0 (bin) / 0 (dec) | 01111111 (bin) / 127 (dec) | 127 (dec) | Passed |
| 1018 | 10111101 (bin) / 189 (dec) | 100 (bin) / 4 (dec) | 0 (bin) / 0 (dec) | 11010000 (bin) / 208 (dec) | 208 (dec) | Passed |
| 1495 | 10111010 (bin) / 186 (dec) | 100 (bin) / 4 (dec) | 0 (bin) / 0 (dec) | 10100000 (bin) / 160 (dec) | 160 (dec) | Passed |
| 242 | 01011110 (bin) / 94 (dec) | 011 (bin) / 3 (dec) | 0 (bin) / 0 (dec) | 11110000 (bin) / 240 (dec) | 240 (dec) | Passed |
| 586 | 00000110 (bin) / 6 (dec) | 011 (bin) / 3 (dec) | 1 (bin) / 1 (dec) | 00000000 (bin) / 0 (dec) | 0 (dec) | Passed |
| 1293 | 10111100 (bin) / 188 (dec) | 000 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 10111100 (bin) / 188 (dec) | 188 (dec) | Passed |
| 1644 | 00000101 (bin) / 5 (dec) | 100 (bin) / 4 (dec) | 0 (bin) / 0 (dec) | 01010000 (bin) / 80 (dec) | 80 (dec) | Passed |
| 12 | 01001101 (bin) / 77 (dec) | 110 (bin) / 6 (dec) | 0 (bin) / 0 (dec) | 01000000 (bin) / 64 (dec) | 64 (dec) | Passed |
| 134 | 00100110 (bin) / 38 (dec) | 000 (bin) / 0 (dec) | 1 (bin) / 1 (dec) | 00100110 (bin) / 38 (dec) | 38 (dec) | Passed |

| Test Case | Input data_in | Input shift_amount | Input direction | Output data_out (Actual) | Expected data_out | Status |
|---|---|---|---|---|---|---|
| 2373 | 10001011 (bin) / 139 (dec) | 111 (bin) / 7 (dec) | 0 (bin) / 0 (dec) | 10000000 (bin) / 128 (dec) | 128 (dec) | Passed |
| 1784 | 00101010 (bin) / 42 (dec) | 101 (bin) / 5 (dec) | 0 (bin) / 0 (dec) | 01000000 (bin) / 64 (dec) | 64 (dec) | Passed |

## Rule: LogicalShifterRule

Input Variables: data_in, shift_amount, direction

Output Variables: data_out

Bit Width: 4

Pattern: StringMatchPattern

```
            def matches(self, filename):
                #print(self.pattern, filename)
                return self.pattern == filename
```

Generate expected values function:

```
            def generate_expected(self, test_case):
                max_val = (1 << self.bit_width) - 1
                if test_case["direction"] == 0:
                    result = (test_case["data_in"] << test_case["shift_amount"]) & max_val
                else:  # right shift
                    result = (test_case["data_in"] >> test_case["shift_amount"]) & max_val

                outs = {
                    "data_out": result
                }
                return outs
```