Scoring Test

Table of Contents

Testbench Summary

Component	Total Tests	Passed	Failed
full_adder	8	8	0

Total tests: 8

Passed tests: 8

Failed tests: 0

Test Case	Input a	Input b	Input cin	Output sum (Actual)	Expected sum	Output cout (Actual)	Expected cout	Status
0	1 (bin) / 1 (dec)	1 (bin) / 1 (dec)	1 (bin) / 1 (dec)	1 (bin) / 1 (dec)	1 (dec)	1 (bin) / 1 (dec)	1 (dec)	Passed
4	1 (bin) / 1 (dec)	1 (bin) / 1 (dec)	1 (bin) / 1 (dec)	1 (bin) / 1 (dec)	1 (dec)	1 (bin) / 1 (dec)	1 (dec)	Passed
6	0 (bin) / 0 (dec)	1 (bin) / 1 (dec)	0 (bin) / 0 (dec)	1 (bin) / 1 (dec)	1 (dec)	0 (bin) / 0 (dec)	0 (dec)	Passed
2	0 (bin) / 0 (dec)	0 (bin) / 0 (dec)	0 (bin) / 0 (dec)	0 (bin) / 0 (dec)	0 (dec)	0 (bin) / 0 (dec)	0 (dec)	Passed
7	0 (bin) / 0 (dec)	1 (bin) / 1 (dec)	0 (bin) / 0 (dec)	1 (bin) / 1 (dec)	1 (dec)	0 (bin) / 0 (dec)	0 (dec)	Passed
3	1 (bin) / 1 (dec)	1 (bin) / 1 (dec)	1 (bin) / 1 (dec)	1 (bin) / 1 (dec)	1 (dec)	1 (bin) / 1 (dec)	1 (dec)	Passed
1	1 (bin) / 1 (dec)	0 (bin) / 0 (dec)	0 (bin) / 0 (dec)	1 (bin) / 1 (dec)	1 (dec)	0 (bin) / 0 (dec)	0 (dec)	Passed
5	1 (bin) / 1 (dec)	0 (bin) / 0 (dec)	1 (bin) / 1 (dec)	0 (bin) / 0 (dec)	0 (dec)	1 (bin) / 1 (dec)	1 (dec)	Passed

Rule: AdderRule

Input Variables: a, b, cin

Output Variables: sum, cout

Bit Width: 8

Pattern: SubstringPattern

```
def matches(self, filename):
    return self.pattern in filename
```

Generate expected values function:

```
def generate_expected(self, test_case):
    max_val = (1 << self.bit_width) - 1
    if "cin" in test_case:
        sum_val = test_case["a"] + test_case["b"] + test_case["cin"]
        outs = {
            "sum": sum_val & max_val,
            "cout": sum_val >> self.bit_width
        }
    else:
        sum_val = test_case["a"] + test_case["b"]
        outs = {
            "sum": sum_val & max_val,
            "cout": sum_val >> self.bit_width
        }
    return outs
```