```verilog
// File: problem_2_5_1_funny_module.v
// John Hubbard, 03 Oct 2014
// hw2 assignment

module funny_module_structural(a, b, c, d, f);
    input a, b, c, d;
    output f;

    wire w[9];
    wire notA, notB, notC, notD;

    not N1(notA, a);
    not N2(notB, b);
    not N3(notC, c);
    not N4(notD, d);

    and A1(w[0], a, b, c, d);
    and A2(w[1], notA, notB, notC, notD);
    and A3(w[2], a, notD);
    and A4(w[3], notA, b, notC);
    and A5(w[4], notB, d);
    and A6(w[5], notA, b, notC);
    and A7(w[6], a, b, notC, d);
    and A8(w[7], b, c, d);
    and A9(w[8], notA, c, notD);

    or O1(f, w[0], w[1], w[2], w[3], w[4], w[5],
          w[6], w[7], w[8]);
endmodule

module funny_module_dataflow(a, b, c, d, f);
    input a, b, c, d;
    output f;

    assign f =
    (a & b & c & d)       |
    (~a & ~b & ~c & ~d)   |
    (a & ~d)              |
    (~a & b & ~c )        |
    (~b & d)              |
    (~a & b & ~c)         |
    (a & b & ~c & d)      |
    (b & c & d)           |
    (~a & c & ~d);
endmodule

module funny_module_simplified_structural(a, b, c, d, f);
    input a, b, c, d;
    output f;

    not F1(f, 0);
endmodule

module funny_module_simplified_dataflow(a, b, c, d, f);
```

```verilog
    input a, b, c, d;
    output f;

    assign f = 1;
endmodule
```