```
// File: minor.v
// John Hubbard, 19 Oct 2014
// hw5a assignment for Verilog 0764 class: Problem 4.4.1: Minority Circuit
//

// Structural Verilog version, as required by the problem statement:

module minor(a, f);
    input [4:0] a;
    output f;

    wire x0;

    // Minority circuit with an odd number of inputs is the precise opposite of
    // a majority circuit:

    major maj(a, x0);
    not f0(f, x0);
endmodule

module major(a, f);
    input [4:0] a;
    output f;

    wire na0, na1, na2, na3, na4;
    wire x0, x1, x2, x3, x4, x5, x6, x7, x8, x9, xa, xb, xc, xd, xe, xf;

    not notA4(na4, a[4]);
    not notA3(na3, a[3]);
    not notA2(na2, a[2]);
    not notA1(na1, a[1]);
    not notA0(na0, a[0]);

    and X0(x0,    na4,  na3,     a[2],    a[1],   a[0]);     // 00111
    and X1(x1,    na4,   a[3],  na2,      a[1],   a[0]);     // 01011
    and X2(x2,    na4,   a[3],   a[2],   na1,     a[0]);     // 01101
    and X3(x3,    na4,   a[3],   a[2],    a[1],  na0);       // 01110

    and X4(x4,    na4, a[3],  a[2],      a[1],      a[0]);   // 01111
    and X5(x5,    a[4], na3,   na2,        a[1],      a[0]); // 10011
    and X6(x6,    a[4], na3,    a[2],     na1,        a[0]); // 10101
    and X7(x7,    a[4], na3,    a[2],      a[1],     na0);   // 10110

    and X8(x8,    a[4], na3,   a[2],     a[1],      a[0]);   // 10111
    and X9(x9,    a[4], a[3],   na2,      na1,      a[0]);   // 11001
    and Xa(xa,    a[4], a[3],   na2,       a[1],     na0);   // 11010
    and Xb(xb,    a[4], a[3],   na2,       a[1],      a[0]); // 11011

    and Xc(xc,    a[4], a[3],   a[2],     na1,     na0);     // 11100
    and Xd(xd,    a[4], a[3],   a[2],     na1,      a[0]);   // 11101
    and Xe(xe,    a[4], a[3],   a[2],      a[1],   na0);     // 11110
    and Xf(xf,    a[4], a[3],   a[2],      a[1],    a[0]);   // 11111

    or f0(f, x0, x1, x2, x3, x4, x5, x6, x7,
```

```
            x8, x9, xa, xb, xc, xd, xe, xf);

endmodule

// Dataflow version of the same logic. This provides a reference, which the
// testbench uses to compare against:
module minor_reference(a, f);
    input [4:0] a;
    output f;

    // Just negate the majority circuit logic, which was already tested in
    // the previous problem assignment:

    assign f = !(
      (  !a[4] & !a[3] & a[2] & a[1] & a[0]  )  |   // 00111
      (  !a[4] & a[3] & !a[2] & a[1] & a[0]  )  |   // 01011
      (  !a[4] & a[3] & a[2] & !a[1] & a[0]  )  |   // 01101
      (  !a[4] & a[3] & a[2] & a[1] & !a[0]  )  |   // 01110

      (  !a[4] & a[3] & a[2] & a[1] & a[0]   )  |   // 01111
      (  a[4] & !a[3] & !a[2] & a[1] & a[0] )  |    // 10011
      (  a[4] & !a[3] & a[2] & !a[1] & a[0]  )  |   // 10101
      (  a[4] & !a[3] & a[2] & a[1] & !a[0]  )  |   // 10110

      (  a[4] & !a[3] & a[2] & a[1] & a[0]   )  |   // 10111
      (  a[4] & a[3] & !a[2] & !a[1] & a[0]  )  |   // 11001
      (  a[4] & a[3] & !a[2] & a[1] & !a[0]  )  |   // 11010
      (  a[4] & a[3] & !a[2] & a[1] & a[0]   )  |   // 11011

      (  a[4] & a[3] & a[2] & !a[1] & !a[0]  )  |   // 11100
      (  a[4] & a[3] & a[2] & !a[1] & a[0]   )  |   // 11101
      (  a[4] & a[3] & a[2] & a[1] & !a[0]   )  |   // 11110
      (  a[4] & a[3] & a[2] & a[1] & a[0]));         // 11111

endmodule
```