```
// File: counter_4bit.v
// John Hubbard, 04 Oct 2014
// hw3 assignment

`timescale 1ns/1ns
module d_flop(d, clk, q, qbar);
    input d, clk;
    output q, qbar;

    reg q, qbar;
    wire d, clk;

    always @(posedge clk)
    begin
        q = d;
        qbar = !q;
    end
endmodule

module counter_4bit(clk, reset, count);
    input clk, reset;
    output [3:0] count;

    wire [3:0] d_in;
    wire [3:0] d_in_and_not_reset;
    wire [3:0] count;
    wire [3:0] count_qbar;
    wire d2_d1bar;
    wire d2_d0bar;
    wire d2bar_d1_d0;
    wire d3_d2bar;
    wire d3_d1bar;
    wire d3_d0bar;
    wire d3bar_d2_d1_d0;
    wire not_reset;

    d_flop d0(d_in_and_not_reset[0], clk, count[0], count_qbar[0]);
    d_flop d1(d_in_and_not_reset[1], clk, count[1], count_qbar[1]);
    d_flop d2(d_in_and_not_reset[2], clk, count[2], count_qbar[2]);
    d_flop d3(d_in_and_not_reset[3], clk, count[3], count_qbar[3]);

    xor bit1_next(d_in[1], count[0], count[1]);

    and bit2_and0(d2_d1bar, count[2], count_qbar[1]);
    and bit2_and1(d2_d0bar, count[2], count_qbar[0]);
    and bit2_and2(d2bar_d1_d0, count_qbar[2], count[1], count[0]);
    or  bit2_or1(d_in[2], d2_d1bar, d2_d0bar, d2bar_d1_d0);

    and bit3_and0(d3_d2bar, count[3], count_qbar[2]);
    and bit3_and1(d3_d1bar, count[3], count_qbar[1]);
    and bit3_and2(d3_d0bar, count[3], count_qbar[0]);
    and bit3_and4(d3bar_d2_d1_d0, count_qbar[3], count[2], count[1], count[0]);
    or bit3_or1(d_in[3], d3_d2bar, d3_d1bar, d3_d0bar, d3bar_d2_d1_d0);
```

```
    not reset_inv(not_reset, reset);
    and f0(d_in_and_not_reset[0], not_reset, count_qbar[0]);
    and f1(d_in_and_not_reset[1], not_reset, d_in[1]);
    and f2(d_in_and_not_reset[2], not_reset, d_in[2]);
    and f3(d_in_and_not_reset[3], not_reset, d_in[3]);
endmodule

/*
Test run:
vlog ../*.v
# Model Technology ModelSim ALTERA vlog 10.1e Compiler 2013.06 Jun 12 2013
# -- Compiling module d_flop
# -- Compiling module counter_4bit
# -- Compiling module counter_4bit_tb
#
# Top level modules:
#   counter_4bit_tb
vsim work.counter_4bit_tb
# vsim work.counter_4bit_tb
# Loading work.counter_4bit_tb
# Loading work.counter_4bit
# Loading work.d_flop
add wave *
run 330 ns
# time                    1 tclk, treset: 0 1, tcount:  x
# time                    5 tclk, treset: 1 1, tcount:  0
# time                   10 tclk, treset: 0 1, tcount:  0
# time                   11 tclk, treset: 0 0, tcount:  0
# time                   15 tclk, treset: 1 0, tcount:  1
# time                   20 tclk, treset: 0 0, tcount:  1
# time                   25 tclk, treset: 1 0, tcount:  2
# time                   30 tclk, treset: 0 0, tcount:  2
# time                   35 tclk, treset: 1 0, tcount:  3
# time                   40 tclk, treset: 0 0, tcount:  3
# time                   45 tclk, treset: 1 0, tcount:  4
# time                   50 tclk, treset: 0 0, tcount:  4
# time                   55 tclk, treset: 1 0, tcount:  5
# time                   60 tclk, treset: 0 0, tcount:  5
# time                   65 tclk, treset: 1 0, tcount:  6
# time                   70 tclk, treset: 0 0, tcount:  6
# time                   75 tclk, treset: 1 0, tcount:  7
# time                   80 tclk, treset: 0 0, tcount:  7
# time                   85 tclk, treset: 1 0, tcount:  8
# time                   90 tclk, treset: 0 0, tcount:  8
# time                   95 tclk, treset: 1 0, tcount:  9
# time                  100 tclk, treset: 0 0, tcount:  9
# time                  105 tclk, treset: 1 0, tcount: 10
# time                  110 tclk, treset: 0 0, tcount: 10
# time                  115 tclk, treset: 1 0, tcount: 11
# time                  120 tclk, treset: 0 0, tcount: 11
# time                  125 tclk, treset: 1 0, tcount: 12
# time                  130 tclk, treset: 0 0, tcount: 12
# time                  135 tclk, treset: 1 0, tcount: 13
# time                  140 tclk, treset: 0 0, tcount: 13
```

```
# time                  145 tclk, treset: 1 0, tcount: 14
# time                  150 tclk, treset: 0 0, tcount: 14
# time                  155 tclk, treset: 1 0, tcount: 15
# time                  160 tclk, treset: 0 0, tcount: 15
# time                  165 tclk, treset: 1 0, tcount:  0
# time                  170 tclk, treset: 0 0, tcount:  0
# time                  175 tclk, treset: 1 0, tcount:  1
# time                  180 tclk, treset: 0 0, tcount:  1
# time                  185 tclk, treset: 1 0, tcount:  2
# time                  190 tclk, treset: 0 0, tcount:  2
# time                  195 tclk, treset: 1 0, tcount:  3
# time                  200 tclk, treset: 0 0, tcount:  3
# time                  205 tclk, treset: 1 0, tcount:  4
# time                  210 tclk, treset: 0 0, tcount:  4
# time                  215 tclk, treset: 1 0, tcount:  5
# time                  220 tclk, treset: 0 0, tcount:  5
# time                  225 tclk, treset: 1 0, tcount:  6
# time                  230 tclk, treset: 0 0, tcount:  6
# time                  235 tclk, treset: 1 0, tcount:  7
# time                  240 tclk, treset: 0 0, tcount:  7
# time                  245 tclk, treset: 1 0, tcount:  8
# time                  250 tclk, treset: 0 0, tcount:  8
# time                  255 tclk, treset: 1 0, tcount:  9
# time                  260 tclk, treset: 0 0, tcount:  9
# time                  265 tclk, treset: 1 0, tcount: 10
# time                  270 tclk, treset: 0 0, tcount: 10
# time                  275 tclk, treset: 1 0, tcount: 11
# time                  280 tclk, treset: 0 0, tcount: 11
# time                  285 tclk, treset: 1 0, tcount: 12
# time                  290 tclk, treset: 0 0, tcount: 12
# time                  295 tclk, treset: 1 0, tcount: 13
# time                  300 tclk, treset: 0 0, tcount: 13
# time                  305 tclk, treset: 1 0, tcount: 14
# time                  310 tclk, treset: 0 0, tcount: 14
# time                  315 tclk, treset: 1 0, tcount: 15
# time                  320 tclk, treset: 0 0, tcount: 15
# time                  325 tclk, treset: 1 0, tcount:  0


*/
```