```vhdl
library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity rd_wr_hit_miss_reg is
    port(
        rd_wr       :   in  std_logic;
        rd_wr_set_en:   in  std_logic;
        hit_miss    :   in  std_logic;
        hit_miss_en :   in  std_logic; --signal for latching hit miss
        clk         :   in  std_logic;
        reset       :   in  std_logic;
        Gnd         :   in  std_logic;
        rd_wr_o     :   out std_logic;
        rd_hit      :   out std_logic;
        wr_hit      :   out std_logic;
        rd_miss     :   out std_logic;
        wr_miss     :   out std_logic
    );
end rd_wr_hit_miss_reg;

architecture structural of rd_wr_hit_miss_reg is

    component dff_reset
        port(
            d       :   in  std_logic;
            clk     :   in  std_logic;
            reset   :   in  std_logic;
            Gnd     :   in  std_logic;
            q       :   out std_logic
        );
    end component;

    component Dlatch_Reset
    port(
        d       :   in  std_logic;
        clk     :   in  std_logic;
        reset   :   in  std_logic;
        Gnd     :   in  std_logic;
        q       :   out std_logic
    );
    end component;

    component and2
        port(
            in1     :   in  std_logic;
            in2     :   in  std_logic;
            out1    :   out std_logic
        );
    end component;

    component and4
        port(
            in1     :   in  std_logic;
            in2     :   in  std_logic;
            in3     :   in  std_logic;
            in4     :   in  std_logic;
            out1    :   out std_logic
        );
    end component;

    component and3
        port(
            in1     :   in  std_logic;
            in2     :   in  std_logic;
            in3     :   in  std_logic;
            out1    :   out std_logic
        );
    end component;

    component or2
        port(
            in1     :   in  std_logic;
            in2     :   in  std_logic;
            out1    :   out std_logic
        );
    end component;

    component invX1
        port(
            in1     :   in  std_logic;
            out1    :   out std_logic
        );
    end component;

    signal n_rd_wr      :   std_logic;
    signal n_hit_miss   :   std_logic;
    signal rhit         :   std_logic;
    signal rmiss        :   std_logic;
    signal whit         :   std_logic;
    signal wmiss        :   std_logic;

    signal reg1_in      :   std_logic;
    signal reg2_in      :   std_logic;

    signal latched_rd_wr    :   std_logic;
    signal latched_hit_miss :   std_logic;
    signal nen              :   std_logic;
    signal clk_en1          :   std_logic;
    signal clk_en2          :   std_logic;

    signal enabled_clk      :   std_logic;

    signal nreset           :   std_logic;

    for inv1, inv2, n_en, nrst  :   invX1 use entity work.invX1(structural);
    for and_1, and_2, and_3, and_4, asdf, reg1in  : and2 use entity work.and2(structural);
    for clk_en, nrsten  :   or2 use entity work.or2(structural);
    for reg1    :   dff_reset use entity work.dff_reset(structural);
    for latch   :   Dlatch_Reset use entity work.Dlatch_Reset(structural);


begin

    rd_wr_o <=  latched_rd_wr;

    inv1    :   invX1       port map(latched_rd_wr, n_rd_wr);
    inv2    :   invX1       port map(latched_hit_miss, n_hit_miss);
    n_en    :   invX1       port map(rd_wr_set_en, nen);
    nrst    :   invX1       port map(reset, nreset);

    and_1   :   and2        port map(latched_rd_wr, latched_hit_miss, rd_hit);
    and_2   :   and2        port map(n_rd_wr, latched_hit_miss, wr_hit);
    and_3   :   and2        port map(latched_rd_wr, n_hit_miss, rd_miss);
    and_4   :   and2        port map(n_rd_wr, n_hit_miss, wr_miss);


    clk_en  :   or2         port map(nen, clk, clk_en1);
    nrsten  :   or2         port map(nreset, clk, clk_en2);
    asdf    :   and2        port map(clk_en1, clk_en2, enabled_clk);

    reg1in  :   and2        port map(rd_wr,  rd_wr_set_en, reg1_in);

    reg1    :   dff_reset   port map(reg1_in, enabled_clk, reset, Gnd, latched_rd_wr);
    latch   :   Dlatch_Reset    port map(hit_miss, hit_miss_en, reset, Gnd, latched_hit_miss);

end structural;
```