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

entity Cache_Cell is
    port(
        Data    :   in  std_logic;
        W_En    :   in  std_logic;
        reset   :   in  std_logic;
        Gnd     :   in  std_logic;
        Output  :   out std_logic;
        Rd_En   :   in  std_logic;
        nRd_En  :   in  std_logic
    );
end Cache_Cell;

architecture structural of Cache_Cell is

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

    component tx
    port(
        sel     :   in  std_logic;
        selnot  :   in  std_logic;
        input   :   in  std_logic;
        output  :   out std_logic
    );
    end component;

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

    signal Dout :   std_logic;
    signal b1   :   std_logic; --buffer intermediate signal
    signal bout :   std_logic; --buffer output
    signal eout :   std_logic; --enable signal going into Dlatch

    for r_or    :   or2 use entity work.or2(structural);
    for dl      :   Dlatch_Reset use entity work.Dlatch_Reset(structural);
    for trans   :   tx  use entity work.tx(structural);

begin

    r_or    :   or2             port map(reset, W_En, eout);
    dl      :   Dlatch_Reset    port map(Data, eout, reset, Gnd, Dout);
    trans   :   tx              port map(Rd_En, nRd_En, Dout, Output);

end structural;
```