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

entity Cache_Block is
    port(
        Data_In     :   in  std_logic_vector(7 downto 0);
        Tag_In      :   in  std_logic_vector(2 downto 0);
        Set_Valid   :   in  std_logic;
        Rd_Wr       :   in  std_logic;
        Cache_Write :   in  std_logic;
        Col_En      :   in  std_logic_vector(3 downto 0);
        Row_En      :   in  std_logic_vector(7 downto 0);
        Tag_Wr_En   :   in  std_logic;
        Gnd         :   in  std_logic;
        reset       :   in  std_logic;
        Data_Out    :   out std_logic_vector(7 downto 0);
        Tag_Out     :   out std_logic_vector(2 downto 0);
        Valid_Out   :   out std_logic
    );
end Cache_Block;

architecture structural of Cache_Block is

    component Cache_Cell_Row
    port(
        Data_In     :   in  std_logic_vector(7 downto 0);
        Tag_In      :   in  std_logic_vector(2 downto 0);
        Set_Valid   :   in  std_logic;
        Col0_Rd_En  :   in  std_logic;
        Col1_Rd_En  :   in  std_logic;
        Col2_Rd_En  :   in  std_logic;
        Col3_Rd_En  :   in  std_logic;
        Col0_Wr_En  :   in  std_logic;
        Col1_Wr_En  :   in  std_logic;
        Col2_Wr_En  :   in  std_logic;
        Col3_Wr_En  :   in  std_logic;
        Tag_Wr_En   :   in  std_logic;
        Row_Rd_En   :   in  std_logic;
        Row_Wr_En   :   in  std_logic;
        Row_En      :   in  std_logic;
        Gnd         :   in  std_logic;
        reset       :   in  std_logic;
        Data_Out    :   out std_logic_vector(7 downto 0);
        Tag_Out     :   out std_logic_vector(2 downto 0);
        Valid_Out   :   out std_logic
    );
    end component;

    component and2
    port(
        in1     :   in  std_logic;
        in2     :   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 Col0_Rd_En   :   std_logic;
    signal Col1_Rd_En   :   std_logic;
    signal Col2_Rd_En   :   std_logic;
    signal Col3_Rd_En   :   std_logic;
    signal Col0_Wr_En   :   std_logic;
    signal Col1_Wr_En   :   std_logic;
    signal Col2_Wr_En   :   std_logic;
    signal Col3_Wr_En   :   std_logic;
    signal nrdwr        :   std_logic;
    signal Write        :   std_logic;

    signal Row_Write    :   std_logic_vector(7 downto 0);
    signal Row_Read     :   std_logic_vector(7 downto 0);
    signal wr_en        :   std_logic;

    for and_0, and_1, and_2, and_3, and_4, and_5, and_6, and_7, r0r, r1r, r2r, r3r, r4r, r5r, r6r, r7r, r0w, r1w, r2w, r3w, r4w, r5w, r6w, r7w : and2 use entity work.and2(structural);
    for row0, row1, row2, row3, row4, row5, row6, row7 : Cache_Cell_Row use entity work.Cache_Cell_Row(structural);

begin


    and_0   :   and2    port map(rd_wr, Col_En(0), Col0_Rd_En);
    and_1   :   and2    port map(rd_wr, Col_En(1), Col1_Rd_En);
    and_2   :   and2    port map(rd_wr, Col_En(2), Col2_Rd_En);
    and_3   :   and2    port map(rd_wr, Col_En(3), Col3_Rd_En);
    and_4   :   and2    port map(Cache_Write, Col_En(0), Col0_Wr_En);
    and_5   :   and2    port map(Cache_Write, Col_En(1), Col1_Wr_En);
    and_6   :   and2    port map(Cache_Write, Col_En(2), Col2_Wr_En);
    and_7   :   and2    port map(Cache_Write, Col_En(3), Col3_Wr_En);

    r0r     :   and2    port map(Row_En(0), rd_wr, Row_Read(0));
    r1r     :   and2    port map(Row_En(1), rd_wr, Row_Read(1));
    r2r     :   and2    port map(Row_En(2), rd_wr, Row_Read(2));
    r3r     :   and2    port map(Row_En(3), rd_wr, Row_Read(3));
    r4r     :   and2    port map(Row_En(4), rd_wr, Row_Read(4));
    r5r     :   and2    port map(Row_En(5), rd_wr, Row_Read(5));
    r6r     :   and2    port map(Row_En(6), rd_wr, Row_Read(6));
    r7r     :   and2    port map(Row_En(7), rd_wr, Row_Read(7));

    r0w     :   and2    port map(Row_En(0), Cache_Write, Row_Write(0));
    r1w     :   and2    port map(Row_En(1), Cache_Write, Row_Write(1));
    r2w     :   and2    port map(Row_En(2), Cache_Write, Row_Write(2));
    r3w     :   and2    port map(Row_En(3), Cache_Write, Row_Write(3));
    r4w     :   and2    port map(Row_En(4), Cache_Write, Row_Write(4));
    r5w     :   and2    port map(Row_En(5), Cache_Write, Row_Write(5));
    r6w     :   and2    port map(Row_En(6), Cache_Write, Row_Write(6));
    r7w     :   and2    port map(Row_En(7), Cache_Write, Row_Write(7));

    row0    :   Cache_Cell_Row  port map(Data_In, Tag_In, Set_Valid, Col0_Rd_E
n, Col1_Rd_En, Col2_Rd_En, Col3_Rd_En, Col0_Wr_En, Col1_Wr_En, Col2_Wr_En, Col3_Wr_en, Tag_Wr_En, Row_Read(0), Row_Write(0), Row_En(0), Gnd, reset, Data_Out, Tag_Out, Valid_Out);
    row1    :   Cache_Cell_Row  port map(Data_In, Tag_In, Set_Valid, Col0_Rd_En, Col1_Rd_En, Col2_Rd_En, Col3_Rd_En, Col0_Wr_En, Col1_Wr_En, Col2_Wr_En, Col3_Wr_en, Tag_Wr_En, Row_Read(1), Row_Write(1), Row_En(1), Gnd, reset, Data_Out, Tag_Out, Valid_Out);
    row2    :   Cache_Cell_Row  port map(Data_In, Tag_In, Set_Valid, Col0_Rd_En, Col1_Rd_En, Col2_Rd_En, Col3_Rd_En, Col0_Wr_En, Col1_Wr_En, Col2_Wr_En, Col3_Wr_en, Tag_Wr_En, Row_Read(2), Row_Write(2), Row_En(2), Gnd, reset, Data_Out, Tag_Out, Valid_Out);
    row3    :   Cache_Cell_Row  port map(Data_In, Tag_In, Set_Valid, Col0_Rd_En, Col1_Rd_En, Col2_Rd_En, Col3_Rd_En, Col0_Wr_En, Col1_Wr_En, Col2_Wr_En, Col3_Wr_en, Tag_Wr_En, Row_Read(3), Row_Write(3), Row_En(3), Gnd, reset, Data_Out, Tag_Out, Valid_Out);
    row4    :   Cache_Cell_Row  port map(Data_In, Tag_In, Set_Valid, Col0_Rd_En, Col1_Rd_En, Col2_Rd_En, Col3_Rd_En, Col0_Wr_En, Col1_Wr_En, Col2_Wr_En, Col3_Wr_en, Tag_Wr_En, Row_Read(4), Row_Write(4), Row_En(4), Gnd, reset, Data_Out, Tag_Out, Valid_Out);
    row5    :   Cache_Cell_Row  port map(Data_In, Tag_In, Set_Valid, Col0_Rd_En, Col1_Rd_En, Col2_Rd_En, Col3_Rd_En, Col0_Wr_En, Col1_Wr_En, Col2_Wr_En, Col3_Wr_en, Tag_Wr_En, Row_Read(5), Row_Write(5), Row_En(5), Gnd, reset, Data_Out, Tag_Out, Valid_Out);
    row6    :   Cache_Cell_Row  port map(Data_In, Tag_In, Set_Valid, Col0_Rd_En, Col1_Rd_En, Col2_Rd_En, Col3_Rd_En, Col0_Wr_En, Col1_Wr_En, Col2_Wr_En, Col3_Wr_en, Tag_Wr_En, Row_Read(6), Row_Write(6), Row_En(6), Gnd, reset, Data_Out, Tag_Out, Valid_Out);
    row7    :   Cache_Cell_Row  port map(Data_In, Tag_In, Set_Valid, Col0_Rd_En, Col1_Rd_En, Col2_Rd_En, Col3_Rd_En, Col0_Wr_En, Col1_Wr_En, Col2_Wr_En, Col3_Wr_en, Tag_Wr_En, Row_Read(7), Row_Write(7), Row_En(7), Gnd, reset, Data_Out, Tag_Out, Valid_Out);
end structural;


```