```
  1: library STD;
  2: library IEEE;
  3: use IEEE.std_logic_1164.all;
  4:
  5: entity Cache_Block is
  6:     port(
  7:         Data_In     :   in  std_logic_vector(7 downto 0);
  8:         Tag_In      :   in  std_logic_vector(2 downto 0);
  9:         Set_Valid   :   in  std_logic;
 10:         Rd_Wr       :   in  std_logic;
 11:         Cache_Write :   in  std_logic;
 12:         Col_En      :   in  std_logic_vector(3 downto 0);
 13:         Row_En      :   in  std_logic_vector(7 downto 0);
 14:         Tag_Wr_En   :   in  std_logic;
 15:         Gnd         :   in  std_logic;
 16:         reset       :   in  std_logic;
 17:         Data_Out    :   out std_logic_vector(7 downto 0);
 18:         Tag_Out     :   out std_logic_vector(2 downto 0);
 19:         Valid_Out   :   out std_logic
 20:     );
 21: end Cache_Block;
 22:
 23: architecture structural of Cache_Block is
 24:
 25:     component Cache_Cell_Row
 26:     port(
 27:         Data_In     :   in  std_logic_vector(7 downto 0);
 28:         Tag_In      :   in  std_logic_vector(2 downto 0);
 29:         Set_Valid   :   in  std_logic;
 30:         Col0_Rd_En  :   in  std_logic;
 31:         Col1_Rd_En  :   in  std_logic;
 32:         Col2_Rd_En  :   in  std_logic;
 33:         Col3_Rd_En  :   in  std_logic;
 34:         Col0_Wr_En  :   in  std_logic;
 35:         Col1_Wr_En  :   in  std_logic;
 36:         Col2_Wr_En  :   in  std_logic;
 37:         Col3_Wr_En  :   in  std_logic;
 38:         Tag_Wr_En   :   in  std_logic;
 39:         Row_Rd_En   :   in  std_logic;
 40:         Row_Wr_En   :   in  std_logic;
 41:         Row_En      :   in  std_logic;
 42:         Gnd         :   in  std_logic;
 43:         reset       :   in  std_logic;
 44:         Data_Out    :   out std_logic_vector(7 downto 0);
 45:         Tag_Out     :   out std_logic_vector(2 downto 0);
 46:         Valid_Out   :   out std_logic
 47:     );
 48:     end component;
 49:
 50:     component and2
 51:     port(
 52:         in1     :   in  std_logic;
 53:         in2     :   in  std_logic;
 54:         out1    :   out std_logic
 55:     );
 56:     end component;
 57:
 58:     component or2
 59:     port(
 60:         in1     :   in  std_logic;
 61:         in2     :   in  std_logic;
 62:         out1    :   out std_logic
 63:     );
 64:     end component;
 65:
 66:     component invX1
 67:     port(
 68:         in1     :   in  std_logic;
 69:         out1    :   out std_logic
 70:     );
 71:     end component;
 72:
 73:     signal Col0_Rd_En   :   std_logic;
 74:     signal Col1_Rd_En   :   std_logic;
 75:     signal Col2_Rd_En   :   std_logic;
 76:     signal Col3_Rd_En   :   std_logic;
 77:     signal Col0_Wr_En   :   std_logic;
 78:     signal Col1_Wr_En   :   std_logic;
 79:     signal Col2_Wr_En   :   std_logic;
 80:     signal Col3_Wr_En   :   std_logic;
 81:     signal nrdwr        :   std_logic;
 82:     signal Write        :   std_logic;
 83:
 84:     signal Row_Write    :   std_logic_vector(7 downto 0);
 85:     signal Row_Read     :   std_logic_vector(7 downto 0);
 86:     signal wr_en        :   std_logic;
 87:
 88:     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 wor
k.and2(structural);
 89:     for row0, row1, row2, row3, row4, row5, row6, row7 : Cache_Cell_Row use en
tity work.Cache_Cell_Row(structural);
 90:
 91: begin
 92:
 93:
 94:     and_0   :   and2    port map(rd_wr, Col_En(0), Col0_Rd_En);
 95:     and_1   :   and2    port map(rd_wr, Col_En(1), Col1_Rd_En);
 96:     and_2   :   and2    port map(rd_wr, Col_En(2), Col2_Rd_En);
 97:     and_3   :   and2    port map(rd_wr, Col_En(3), Col3_Rd_En);
 98:     and_4   :   and2    port map(Cache_Write, Col_En(0), Col0_Wr_En);
 99:     and_5   :   and2    port map(Cache_Write, Col_En(1), Col1_Wr_En);
100:     and_6   :   and2    port map(Cache_Write, Col_En(2), Col2_Wr_En);
101:     and_7   :   and2    port map(Cache_Write, Col_En(3), Col3_Wr_En);
102:
103:     r0r     :   and2    port map(Row_En(0), rd_wr, Row_Read(0));
104:     r1r     :   and2    port map(Row_En(1), rd_wr, Row_Read(1));
105:     r2r     :   and2    port map(Row_En(2), rd_wr, Row_Read(2));
106:     r3r     :   and2    port map(Row_En(3), rd_wr, Row_Read(3));
107:     r4r     :   and2    port map(Row_En(4), rd_wr, Row_Read(4));
108:     r5r     :   and2    port map(Row_En(5), rd_wr, Row_Read(5));
109:     r6r     :   and2    port map(Row_En(6), rd_wr, Row_Read(6));
110:     r7r     :   and2    port map(Row_En(7), rd_wr, Row_Read(7));
111:
112:     r0w     :   and2    port map(Row_En(0), Cache_Write, Row_Write(0));
113:     r1w     :   and2    port map(Row_En(1), Cache_Write, Row_Write(1));
114:     r2w     :   and2    port map(Row_En(2), Cache_Write, Row_Write(2));
115:     r3w     :   and2    port map(Row_En(3), Cache_Write, Row_Write(3));
116:     r4w     :   and2    port map(Row_En(4), Cache_Write, Row_Write(4));
117:     r5w     :   and2    port map(Row_En(5), Cache_Write, Row_Write(5));
118:     r6w     :   and2    port map(Row_En(6), Cache_Write, Row_Write(6));
119:     r7w     :   and2    port map(Row_En(7), Cache_Write, Row_Write(7));
120:
121:     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);
  122:     row1    :   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(1), Row_Write(1), Row_En(1), Gnd, reset, Data_Out, Tag_Out, Valid
_Out);
  123:     row2    :   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(2), Row_Write(2), Row_En(2), Gnd, reset, Data_Out, Tag_Out, Valid
_Out);
  124:     row3    :   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(3), Row_Write(3), Row_En(3), Gnd, reset, Data_Out, Tag_Out, Valid
_Out);
  125:     row4    :   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(4), Row_Write(4), Row_En(4), Gnd, reset, Data_Out, Tag_Out, Valid
_Out);
  126:     row5    :   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(5), Row_Write(5), Row_En(5), Gnd, reset, Data_Out, Tag_Out, Valid
_Out);
  127:     row6    :   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(6), Row_Write(6), Row_En(6), Gnd, reset, Data_Out, Tag_Out, Valid
_Out);
  128:     row7    :   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(7), Row_Write(7), Row_En(7), Gnd, reset, Data_Out, Tag_Out, Valid
_Out);
  129: end structural;
  130:
  131:
```