```
1   ----------------------------------------------------------------------------
2   -- Hardware AMBA bus IP for image scaling
3   ----------------------------------------------------------------------------
4   -- Entity:      TFG_ROW_BUFFER
5   -- File:        TFG_ROW_BUFFER.vhd
6   -- Author:      Luis Miguel Gonzalez Berrocal
7   ----------------------------------------------------------------------------
8   -- VHDL Standard: VHDL'93
9   ----------------------------------------------------------------------------
10  -- Naming Conventions
11  -- active low signals:                  "*_n"
12  -- clock signals:                       "clk", "clk_div#", "clk_#x"
13  -- reset signals:                       "rst", "rst_n"
14  -- generics:                            "C_*"
15  -- user defined types:                  "*_TYPE"
16  -- state machine next state:            "*_ns"
17  -- state machine current state:         "*_cs"
18  -- combinatorial signals:               "*_com"
19  -- pipelined or register delay signals: "*_d#"
20  -- counter signals:                     "*_cnt*"
21  -- clock enable signals:                "*_ce"
22  -- internal version of output port:     "*_i"
23  -- device pins:                         "*_pin"
24  -- ports:                               "Names begin with Uppercase"
25  -- processes:                           "*_PROCESS"
26  -- component instantiations:            "<ENTITY_>I_<#|FUNC>"
27  -- registers:                           "*_REG"
28  ----------------------------------------------------------------------------
29
30  library ieee;
31  use ieee.std_logic_1164.all;
32  use ieee.std_logic_unsigned.all;
33  use ieee.std_logic_arith.all;
34
35  entity TFG_ROW_BUFFER is
36        generic (SIZE : integer := 11;
37                 WORD : integer := 32);
38        port (Clk          : in  std_logic;                                  -- clk
39              Rst_n        : in  std_logic;                                  -- active low
    reset
40              Ram_enable   : in std_logic;                                   -- ram enable
41              Write_enable : in std_logic;                                   -- write enable
42              Address      : in std_logic_vector (SIZE - 1 downto 0);        -- address
43              Data_in      : in std_logic_vector (WORD - 1 downto 0);        -- data in
44              Data_out     : out std_logic_vector (WORD - 1 downto 0));      -- data out
45  end TFG_ROW_BUFFER;
46
47  architecture Behavioral of TFG_ROW_BUFFER is
48
49        type ram_TYPE is array (0 to 2**SIZE - 1) of std_logic_vector (WORD - 1 downto 0
    );
50        signal ram : ram_TYPE := (others => X"CAFECAFE");
51
52  begin
53
54        ram_PROCESS : process (Clk, Rst_n)
55        begin
```

```
56              if (Rst_n = '0') then
57                    Data_out <= (others => '0');
58              elsif (Clk'event and Clk = '1') then
59                    if (Ram_enable = '1') then
60                          if (Write_enable = '1') then
61                                ram (conv_integer (Address)) <= Data_in;
62                          else
63                                Data_out <= ram (conv_integer (Address));
64                          end if;
65                    end if;
66              end if;
67        end process;
68
69  end Behavioral;
70
71
```