```
library ieee;

use ieee.std_logic_1164.all;
use ieee.numeric_std.all;


entity data_memory is
      port(
            mem_rst: in STD_LOGIC;
            mem_write: in STD_LOGIC;
            mem_read: in STD_LOGIC;
            mem_enable: in STD_LOGIC;
            address: in STD_LOGIC_VECTOR(31 downto 0);
            write_data: in STD_LOGIC_VECTOR(31 downto 0);
            read_data: out STD_LOGIC_VECTOR(31 downto 0)
      );
end data_memory;

architecture behavioral of data_memory is
type mem_array is array(0 to (2**8)-1) of STD_LOGIC_VECTOR(31
downto 0);
signal mem_unit: mem_array := (others => (others => '0'));
begin
      process(mem_rst,mem_enable,mem_write,address,write_data)
      begin
                  if(mem_rst = '1') then
                        mem_unit<= (others => (others => '0'));
                  elsif(mem_enable = '1') then
                        if(mem_write = '1') then
                              mem_unit(to_integer(unsigned(address(7
downto 0))))<=write_data;
                        end if;
                  end if;
      end process;
       read_data<=mem_unit(to_integer(unsigned(address(7 downto
0)))) when(mem_read ='1') else (others => '0');
end behavioral;
```