```
--
-- MIPs Processor Developement
-- Eric W. Johnson
-- Valparaiso University
--
--

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
use work.mips_types.all;

--
-- Copyright Jay Brockman, Feb 1997
--  Updated Eric W. Johnson, Feb 1998
--
-- Modified June 2011 Jeffrey Will

-- memory Entity Description
entity dmemory is
   port(
      DIN: in mips_data;
      ADDR: in mips_address;
      DOUT: out mips_data;
      WE, RE: in std_logic;
      CLK: in std_logic

   );
end dmemory;

-- memory Architecture Description
architecture rtl of dmemory is
   subtype ramword is bit_vector(31 DOWNTO 0);
   type rammemory is array (0 to 4096) of ramword;

   --**************************************************
   -- Students: This is where you modify the contents of
   -- Data memory.
   --**************************************************
   signal ram : rammemory := (
            x"00000001", -- 00 through 03
                x"00000002", -- 04 through 07
                x"00000003", --
                x"00000004", --
                x"00000000", --
                others => x"00000000");
begin

   read_Process: process(RE, ADDR)
      variable raddr1 : integer range 0 to 4096;
      variable tempdata : ramword;
   begin
      -- convert address to integer
  IF ( RE = '1' ) THEN
      raddr1 := conv_Integer(ADDR);
      raddr1 := raddr1/4;
      tempdata  := (ram(raddr1));
      DOUT  <= to_stdlogicvector(tempdata);
  END IF;
   end process read_Process;

   write_Process: process(WE, CLK)
      variable waddr : integer range 0 to 4096;
   begin
      if ( WE = '1' AND CLK'EVENT AND CLK = '1' ) then
    -- convert address to integer
    waddr := conv_Integer(ADDR);
        waddr := waddr/4;
    ram(waddr) <= to_bitvector(DIN);
      end if;
   end process write_Process;
end rtl;
```