```
--
-- Copyright Jay Brockman
--
-- MIPs Processor Developement
-- Eric W. Johnson
-- Valparaiso University
--
--
-- Updated Jeffrey Will

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;


-- memory Entity Description
entity imemory is
   port(
      ADDR: in mips_address;
      DOUT: out mips_data
   );
end imemory;

-- memory Architecture Description
architecture rtl of imemory is
   subtype ramword is bit_vector(31 DOWNTO 0);
   type rammemory is array (0 to 1024) of ramword;

   --**************************************************
   -- Students:  You will have to hand-assemble the instructions
   -- Given in the assignment, and fill in the values
   -- below.  (It is like you are "flashing" the instruction
   -- memory with your assembly code
   --**************************************************

   signal ram : rammemory := (
                x"00000820", -- 00: add $1,$0,$0
                x"8C220000", -- 01: lw $2,0($1)
                x"8C230004", -- 02:
                x"00432024", -- 03:
                x"10800001", -- 04:
                x"AC240008", -- 05:
                x"AC24000C", -- 06:
                others => x"00000000");
begin

   read_Process: process(ram, ADDR)
      variable raddr1 : integer range 0 to 1024;
      variable tempdata : ramword;
   begin
      -- convert address to integer
      raddr1 := conv_Integer(ADDR);
      raddr1 := raddr1/4;
      tempdata  := (ram(raddr1));
      DOUT  <= to_stdlogicvector(tempdata);
   end process read_Process;

end rtl;
```