Problem

a) Design and implement a generic de-multiplexor (demux_gen) with m-bit select, i.e., 2^m outputs

```
entity demux_gen is
generic (m : natural);
port (x : in std_logic;
     z : out std_logic_vector(2**m - 1 downto 0);
    sel: in std_logic_vector(m-1 downto 0)
     );
end demux_gen;
```

The device connects the input x to output z(j) where j is equal to the m-bit number sel. It outputs other outputs to zero. Verify the correctness for m = 2.

b) Implement an 8-output demux by connecting two 4-output demux's and one 2-output demux.

```
entity demux8 is
port (x : in std_logic;
     z : out std_logic_vector(7 downto 0);
    sel: in std_logic_vector(2 downto 0)
     );
end demux8;
```

```
----------------------------------------------------
-- Company: Drexel ECE
-- Engineer: Prawat
-- a generic de-multiplexor with m-bit select
-- Test generic component on an 8-output demux
-- two 4-output demux's and one 2-output demux
----------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity demux_gen is
generic (m : natural);
port (x : in std_logic;
      z : out std_logic_vector(2**m - 1 downto 0);
     sel: in std_logic_vector(m-1 downto 0)
      );
end demux_gen;

architecture Behavioral of demux_gen is

begin
process(x, sel)
subtype my_int is integer range 0 to 2**m - 1;
variable j : my_int;
begin

j := 0;
for i in 0 to m-1 loop
   if sel(i) = '1' then j := j + 2**i; end if;
end loop;

for i in 0 to 2**m - 1 loop
  if i=j then z(i) <= x; else
  z(i) <= '0';
  end if;
end loop;
```

```
for i in 0 to 2**m - 1 loop
  if i=j then z(i) <= '1'; else
  z(i) <= '0';
  end if;
end loop;
end if;
end process;
end Behavioral;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity dec328_2x224 is
port (x : in   std_logic_vector(2 downto 0);
       z:  out std_logic_vector(7 downto 0)
      );
end dec328_2x224;

architecture struc of dec328_2x224 is
component decg
generic(m : natural);
port (x : in   std_logic_vector(m-1 downto 0);
        en: in   std_logic;
         z:  out std_logic_vector(2**m - 1 downto 0)
      );
end component;
signal w : std_logic;
begin
w <= not x(2);
u1: decg generic map (2) port map
(x => x(1 downto 0), en => x(2), z => z(7 downto 4));
u2: decg generic map (2) port map
(x => x(1 downto 0), en => w, z => z(3 downto 0));
end struc;
```