-- Multiplexer

-- lab 6

library IEEE; use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity Mux IS

port (

x: in STD\_LOGIC\_Vector(1 downto 0);

a: in STD\_LOGIC;

b: in STD\_LOGIC;

c: in STD\_LOGIC;

d: in STD\_LOGIC;

y: out STD\_LOGIC

);

end Mux;

architecture behavior of Mux is

begin

P1: process(x)

begin

if x="00" then

y<=a;

elsif x="01" then

y<=b;

elsif x="10" then

y<=c;

elsif x="11" then

y<=d;

else y<=a;

end if;

end process P1;

end behavior;

Part 2

-- Decoder

-- lab 6

library IEEE; use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity Decoder IS

port (

x: in STD\_LOGIC\_Vector(2 downto 0);

y: out STD\_LOGIC\_Vector(7 downto 0)

);

end Decoder;

architecture behavior of Decoder is

begin

P1: process(x)

begin

if x="000" then

y<="00000001";

elsif x="001" then

y<="00000010";

elsif x="010" then

y<="00000100";

elsif x="011" then

y<="00001000";

elsif x="100" then

y<="00010000";

elsif x="101" then

y<="00100000";

elsif x="110" then

y<="01000000";

elsif x="111" then

y<="10000000";

else y<="00000001";

end if;

end process P1;

end behavior;

Part 3

-- Encoder

-- lab 6

library IEEE; use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity Encoder IS

port (

x: in STD\_LOGIC\_Vector(2 downto 0);

y: out STD\_LOGIC\_Vector(1 downto 0)

);

end Encoder;

architecture behavior of Encoder is

begin

P1: process(x)

begin

if x(2)='1' then

y<="11";

elsif x(1)='1' then

y<="10";

elsif x(0)='1' then

y<="01";

elsif x(0)='0' then

y<="00";

else y<="00";

end if;

end process P1;

end behavior;