Generator

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity Generator is

generic(N: integer:= 50000000;

DC: integer:= 25000000);

port(

clkin: in std\_logic;

clkout: buffer std\_logic:='1');

end;

architecture one of Generator is

signal count: integer range 0 to N := 0;

begin

process(clkin)

begin

if (clkin 'event and clkin = '1') then

if (count = N) then

count <= 0;

clkout <= '1';

elsif (count = DC) then

count <= count + 1;

clkout <= '0';

else

count <= count + 1;

end if;

end if;

end process;

end;

Parity

library ieee;

use ieee.std\_logic\_1164.all;

entity Parity\_Gen is

generic (N: integer:=4);

port(

inpt: std\_logic\_vector(N-1 downto 0);

outpt: out std\_logic);

end;

architecture one of Parity\_Gen is

begin

process(inpt)

variable x: std\_logic;

begin

x:='0';

for i in 0 to N-1 loop

x := x xor inpt(i);

end loop;

outpt <= x;

end process;

end;

Piso

library ieee;

use ieee.std\_logic\_1164.all;

entity PISO is

generic (n:integer:=4);

port(data: std\_logic\_vector (n-1 downto 0);

clk,sel: in std\_logic;

Q,txe:out std\_logic);

end;

architecture one of PISO is

signal sq: std\_logic\_vector (n-1 downto 0);

signal ntxe: std\_logic;

begin

process(clk)

begin

for i in 0 to n-1 loop

if (clk='1' and clk'event) then

if (sel='0') then sq<=data;

else sq<='0'&sq (n-1 downto 1);

end if;

end if;

end loop;

end process;

q<=sq(0);

txe<= not ntxe;

process(sq)

variable x: std\_logic;

begin

x:='0';

for i in 0 to n-1 loop

x:= x or sq(i);

end loop;

ntxe<=x;

end process;

end;

Transmitter

library ieee;

use ieee.std\_logic\_1164.all;

entity transmiter is

port(data: in std\_logic\_vector (7 downto 0);

clk,parity\_type,go: in std\_logic;

txd, txrdy, txe: out std\_logic);

end;

architecture one of transmiter is

component generator

generic(N: integer:= 50000000;

DC: integer:= 25000000);

port(

clkin: in std\_logic;

clkout: buffer std\_logic:='1');

end component;

component piso

generic (n:integer:=4);

port(data: std\_logic\_vector (n-1 downto 0);

clk,sel: in std\_logic;

Q,txe:out std\_logic);

end component;

component parity\_gen

generic (N: integer:=4);

port(

inpt: std\_logic\_vector(N-1 downto 0);

outpt: out std\_logic);

end component;

begin

end;