```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date:    14:03:45 02/23/2015
-- Design Name:
-- Module Name:    Ethernet - Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Ethernet is
    Port ( RBYTEP : out  STD_LOGIC;
           RCLEANP : out  STD_LOGIC;
           RCVNGP : out  STD_LOGIC;
           RDATAO : out  STD_LOGIC_VECTOR (7 downto 0);
           RDATAI : in  STD_LOGIC_VECTOR (7 downto 0);
           RDONEP : out  STD_LOGIC;
           RENABP : in  STD_LOGIC;
           RSMATIP : out  STD_LOGIC;
           RSTARTP : out  STD_LOGIC;
           RESETN : in  STD_LOGIC;
           CLK : in  STD_LOGIC;
                     TSOCOLP : out  STD_LOGIC;
           TABORTP : in  STD_LOGIC;
           TAVAILP : in  STD_LOGIC;
           TRNSMTP : out  STD_LOGIC;
           TDATAI : in  STD_LOGIC_VECTOR (7 downto 0);
           TDATAO : out  STD_LOGIC_VECTOR (7 downto 0);
           TDONEP : out  STD_LOGIC;
           TLASTP : in  STD_LOGIC;
           TREADDP : out  STD_LOGIC;
           TSTARTP : out  STD_LOGIC;
           TBACKOFF : out  STD_LOGIC;
           TSMCOLP : out  STD_LOGIC);
end Ethernet;


architecture Behavioral of Ethernet is
        --VARIABLES--
        constant SFD: STD_LOGIC_VECTOR(7 downto 0):="10101011";
        constant NOADDRI: STD_LOGIC_VECTOR(47 downto 0):=X"AABBCCDDEEFF";
        signal RCVNGP_aux : STD_LOGIC;
        signal TSOCOLP_aux : STD_LOGIC;
        signal TCOLPREV : STD_LOGIC;
        signal TSMCOLP_aux : STD_LOGIC;
        signal TRNSMTP_aux : STD_LOGIC;
        signal COLLISION_FLAG : STD_LOGIC;
        signal LFSR_REG : STD_LOGIC_VECTOR(24 downto 0);
```



```
        signal BACKOFF_T : STD_LOGIC_VECTOR(24 downto 0);
        subtype HUIT_INT IS INTEGER RANGE 0 TO 8;
        subtype CINQ_INT IS INTEGER RANGE 0 TO 5;
        constant ALTERNATIVE_SEQUENCE: STD_LOGIC_VECTOR(7 downto 0):="10101010";

begin

        reception : process -- reception
                variable compteur : HUIT_INT;
                variable compteurAddr : CINQ_INT;
                variable error_check_add : BOOLEAN;
        begin
                wait until CLK'event and CLK = '1';

                -- Pour les pulses ils sont par défaut à 0
                RCLEANP <= '0';
                RSTARTP <= '0';
                RBYTEP <= '0';
                RDONEP <= '0';

                if RESETN='0' then
                        compteur := 7;
                        compteurAddr := 0;
                        RCVNGP_aux<='0';
                        error_check_add:=FALSE;
                        RSMATIP <= '0';
                        RDATAO <= X"00";
                else
                        if RENABP='1' then
                                compteur := compteur + 1;

                                --Tous les 8 clock on lit un octet
                                if (compteur = 8) then -- octet reçu
                                        --remise du compteur de top d'horloge à 0
                                        compteur:=0;

                                        if (RCVNGP_aux='0') then
                                                --
                                                --              CHECK SFD
                                                --
                                                -- Si on est pas en mode réception on check le SFD
                                                if (RDATAI = SFD) then -- Si premier octet = SFD
                                                        RCVNGP_aux <= '1'; -- Reception d'une trame
                                                        RSTARTP <= '1'; -- Indique au receveur de
réserver des buffers
                                                end if ;

                                        else

                                                --
                                                --              RECEIVING
                                                --
                                                -- Si on est en mode réception on a deux cas
                                                -- Soit on lit l'@ MAC DST ie compteurAddr < 6

                                                --
                                                --              CHECK ADDRESSE
                                                --
                                                if (compteurAddr=0) then
                                                        if (NOADDRI(47 downto 40)=RDATAI) then
                                                                compteurAddr := compteurAddr+1;
                                                        else
                                                                error_check_add:=TRUE;
                                                        end if;
                                                elsif (compteurAddr=1) then
                                                        if (NOADDRI(39 downto 32)=RDATAI) then
compteurAddr := compteurAddr+1;
                                                        else
                                                                error_check_add:=TRUE;
                                                        end if;
                                                elsif (compteurAddr=2) then
                                                        if (NOADDRI(31 downto 24)=RDATAI) then
```

```
compteurAddr := compteurAddr+1;
                                                        else
                                                                error_check_add:=TRUE;
                                                        end if;
                                                elsif (compteurAddr=3) then
                                                        if (NOADDRI(23 downto 16)=RDATAI) then
compteurAddr := compteurAddr+1;
                                                        else
                                                                error_check_add:=TRUE;
                                                        end if;
                                                elsif (compteurAddr=4) then
                                                        if (NOADDRI(15 downto 8)=RDATAI) then
compteurAddr := compteurAddr+1;
                                                        else
                                                                error_check_add:=TRUE;
                                                        end if;
                                                elsif (compteurAddr=5) then
                                                        if (NOADDRI(7 downto 0)=RDATAI) then
                                                                compteurAddr := compteurAddr+1;
                                                                RSMATIP <= '1';
                                                        else
                                                                error_check_add:=TRUE;
                                                        end if;
                                                else

                                                        --
                                                        --              CHECK EFD
                                                        --                                      ||  FORWARD
DATA TO UPPER LAYER
                                                        --
                                                        -- Soit on transmet les données ie
compteurAddr >= 6
                                                        -- Soit on reçoit le end frame delimiter
                                                        if (RDATAI=SFD) then -- (/!\ EFD=SFD)
                                                                --on a fini la reception
                                                                RDONEP <= '1';
                                                                RCVNGP_aux <= '0';
                                                                RSMATIP <= '0';
                                                                compteurAddr:=0;
                                                        else
                                                                --on fait remonter les données
                                                                RDATAO <= RDATAI;
                                                                RBYTEP <= '1';
                                                        end if;
                                                end if;

                                        end if;

                                        --Si on a eu une erreur, il faut reset le compteur d'add et le
compteur
                                        if (error_check_add=TRUE) then
                                                compteur := 7;
                                                compteurAddr:=0;
                                                RCLEANP <= '1';
                                                RCVNGP_aux <= '0';
                                                error_check_add:=FALSE;
                                        end if;
                                end if;
                        end if ;
                end if;
        end process;

        RCVNGP <= RCVNGP_aux;


        transmission : process -- transmission
                variable compteur : HUIT_INT;
                variable compteurAddr : CINQ_INT;
                variable error_check_add : BOOLEAN;
                variable compteur_step : CINQ_INT;
                variable compteur_bit_collision : CINQ_INT;
                variable nb_coll_conseq : INTEGER;
```



```
        begin
                wait until CLK'event and CLK = '1';

                TBACKOFF <= '0';

                -- Pour les pulses ils sont par défaut à 0
                TSTARTP <= '0';
           TREADDP <= '0';
                TDONEP <= '0';

                if RESETN='0' then
                        compteur := 7;
                        compteurAddr := 0;
                        error_check_add:=FALSE;
                        compteur_step :=0;
                        compteur_bit_collision:=0;
                        BACKOFF_T <= "000000000000000000000000";
                        nb_coll_conseq := 0;
                        TRNSMTP_aux <= '0';
                        TDATAO <= X"00";
                else
                        if nb_coll_conseq >= 1 then
                                TCOLPREV <= '1';
                        else
                                TCOLPREV <= '0';
                        end if;
                        if BACKOFF_T /= "000000000000000000000000" then
                                --On est en backoff, il faut donc attendre
                                BACKOFF_T <= BACKOFF_T - 1;
                                TBACKOFF <= '1';
                        elsif TABORTP='1' or TSOCOLP_aux = '1' or TSMCOLP_aux = '1' or COLLISION_FLAG
= '1' then
                                -- reset des variables pour la transmission normale
                                compteurAddr := 0;
                                error_check_add:=FALSE;
                                compteur_step :=0;
                                compteur := compteur +1;
                                --Tous les 8 clock on lit un octet
                                if (compteur = 8) then -- octet reçu
                                        --remise du compteur de top d'horloge à 0
                                        compteur:=0;
                                        if compteur_bit_collision < 4 then
                                                COLLISION_FLAG <= '1';
                                                TDATAO <= ALTERNATIVE_SEQUENCE;
                                                compteur_bit_collision := compteur_bit_collision +1;
                                        else
                                                COLLISION_FLAG <= '0';
                                                TRNSMTP_aux <= '0';
                                                TDONEP <= '1';--Terminaison de la transmission par un
TABORT
                                                compteur_bit_collision := 0;
                                                compteur:=7; --On remet le compteur à 7 pour qu'on
puisse retransmettre directement après le backoff
                                                --On passe en backoff maintenant
                                                --BACKOFF_T <= x"0000" & LFSR_REG( 8 downto 0);
                                                --test Collision multiple pour aller plus vite
                                                BACKOFF_T <= x"00000" & LFSR_REG( 4 downto 0);
                                                --BACKOFF_T <= x"00000" & "00010"; --test TABORT pour
aller plus vite
                                                TBACKOFF <= '1';
                                                --On incrémente le nombre de collision
                                                nb_coll_conseq := nb_coll_conseq + 1;
                                                --Plus de data à transmettre
                                                TDATAO <= X"00";
                                        end if;
                                end if;
                        elsif TAVAILP='1' then
                                --Si on est dans ce process c'est qu'on n'est pas en collision
                                compteur_bit_collision :=0;

                                compteur := compteur +1;
                                --Tous les 8 clock on lit un octet
```

```
                if (compteur = 8) then -- octet reçu

                        --remise du compteur de top d'horloge à 0
                        compteur:=0;
                        TRNSMTP_aux <= '1';

                        if (compteur_step =0) then
                                --
                                --              SEND SFD
                                --
                                -- Première étape : send le SFD
                                TDATAO <= SFD;
                                compteur_step := compteur_step +1;
                                TSTARTP <= '1';
                        elsif(compteur_step =1) then
                                --
                                --              SEND ADD DEST
                                --
                                if(compteurAddr = 0) then --envoie du premier octet de
l'adresse destination

                                        compteurAddr := compteurAddr +1;
                                        TDATAO <= TDATAI ;
                                        TREADDP <= '1';

                                elsif(compteurAddr = 1) then
                                        compteurAddr := compteurAddr +1;
                                        TDATAO <= TDATAI ;
                                        TREADDP <= '1';

                                elsif(compteurAddr = 2) then
                                        compteurAddr := compteurAddr +1;
                                        TDATAO <= TDATAI ;
                                        TREADDP <= '1';

                                elsif(compteurAddr = 3) then
                                        compteurAddr := compteurAddr +1;
                                        TDATAO <= TDATAI ;
                                        TREADDP <= '1';

                                elsif(compteurAddr = 4) then
                                        compteurAddr := compteurAddr +1;
                                        TDATAO <= TDATAI ;
                                        TREADDP <= '1';

                                elsif(compteurAddr = 5) then --envoie du dernier octet
de l'adresse destination

                                        compteurAddr := 0;
                                        TDATAO <= TDATAI ;
                                        TREADDP <= '1';
                                        compteur_step := compteur_step + 1;
                                end if;

                        elsif(compteur_step =2) then
                                --
                                --              SEND ADD SRC
                                --
                                if(compteurAddr=0) then
                                        compteurAddr := compteurAddr+1;
                                        TDATAO <= NOADDRI(47 downto 40);
                                        TREADDP <= '1';
                                elsif(compteurAddr=1) then
                                        compteurAddr := compteurAddr+1;
                                        TDATAO <= NOADDRI(39 downto 32);
                                        TREADDP <= '1';
                                elsif(compteurAddr=2) then
                                        compteurAddr := compteurAddr+1;
                                        TDATAO <= NOADDRI(31 downto 24);
                                        TREADDP <= '1';
                                elsif(compteurAddr=3) then
                                        compteurAddr := compteurAddr+1;
                                        TDATAO <= NOADDRI(23 downto 16);
                                        TREADDP <= '1';
```



```
                                elsif(compteurAddr=4) then
                                        compteurAddr := compteurAddr+1;
                                        TDATAO <= NOADDRI(15 downto 8);
                                        TREADDP <= '1';
                                elsif(compteurAddr=5) then
                                        compteurAddr := 0;
                                        TDATAO <= NOADDRI(7 downto 0);
                                        TREADDP <= '1';
                                        compteur_step := compteur_step+1;
                                end if;

                        elsif(compteur_step =3) then
                                --
                                --              SEND DATA
                                --
                                TDATAO <= TDATAI ;
                                TREADDP <= '1';
                                if(TLASTP='1') then
                                        compteur_step := compteur_step+1;
                                end if;
                        elsif(compteur_step =4) then
                                --
                                --              SEND EFD
                                --
                                TDATAO <= SFD;
                                --RESET DES STEPS
                                compteur_step := 0;
                                TDONEP <= '1';
                                TRNSMTP_aux <= '0';
                                --Tout s'est bien passé on remet le nombre de
collision à 0

                                nb_coll_conseq := 0;
                        end if;

                                end if;
                        else
                                --Si on est ici, par défaut on met 0 en sortie
                                --Puisqu'on n'est ni en transmission ni en collision
                                TDATAO <= X"00";
                        end if;
                end if;

        end process;

        TRNSMTP <= TRNSMTP_aux;

        collision : process -- collision
        begin
                wait until CLK'event and CLK = '1';

                if RCVNGP_aux = '1' and TRNSMTP_aux = '1' and TCOLPREV = '1' then
                        TSMCOLP_aux <= '1';
                        TSOCOLP_aux <= '0';
                elsif RCVNGP_aux = '1' and TRNSMTP_aux = '1' then
                        TSOCOLP_aux <= '1';
                        TSMCOLP_aux <= '0';
                else
                        TSMCOLP_aux <= '0';
                        TSOCOLP_aux <= '0';
                end if;
        end process;

        TSOCOLP <= TSOCOLP_aux;
        TSMCOLP <= TSMCOLP_aux;

        lfsr_proc : process
                variable suivant : STD_LOGIC;
                variable i: integer;
                variable lfsr: std_logic_vector(24 downto 0);
        begin

                wait until CLK'event and CLK = '1';
```

```
                if RESETN='0' then
                        lfsr := "10100000000000000000000100";
                else

                        --On réalise s = ((((R(0)+R(2))+R(3))+R(4))+R(6))
                        suivant := lfsr(0);
                        suivant := suivant xor lfsr(2);
                        suivant := suivant xor lfsr(3);
                        suivant := suivant xor lfsr(4);
                        suivant := suivant xor lfsr(6);

                        --Décalage à droite
                        FOR i IN 24 DOWNTO 1 LOOP
                                lfsr(25-i-1) := lfsr(25-i);
                        END LOOP;

                        --Mise en place du bit suivant
                        lfsr(24) := suivant;
                end if;

                LFSR_REG <= lfsr;
        end process;

end Behavioral;
```