```
----------------------------------------------------------------------------------
-- Company: University of the Ryukyus
-- Engineer: Tomohisa Wada
--
-- Create Date:   10:46:47 01/23/2018
-- Design Name:
-- Module Name:   NN - 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;

-- 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 NN2019_BP is
    Port ( RESET  : in  STD_LOGIC;
           SYSCLK : in  STD_LOGIC;
           COUNT2 : out STD_LOGIC_VECTOR ( 1 downto 0);
           A41    : out STD_LOGIC_VECTOR (23 downto 0);
           A42    : out STD_LOGIC_VECTOR (23 downto 0)
                        );
end NN2019_BP;

architecture Behavioral of NN2019_BP is
```

```
    COMPONENT MUL24
    PORT(
         XX  : in  STD_LOGIC_VECTOR (23 downto 0);
         YY  : in  STD_LOGIC_VECTOR (23 downto 0);
         ZZ  : out STD_LOGIC_VECTOR (23 downto 0);
                           ZZB : in  STD_LOGIC_VECTOR (23 downto 0);
                           XXB : out STD_LOGIC_VECTOR (23 downto 0);
                           YYB : out STD_LOGIC_VECTOR (23 downto 0)
        );
    END COMPONENT;
           COMPONENT RELU_FUNC
           PORT(
                XX  : in  STD_LOGIC_VECTOR (23 downto 0);
         YY  : out STD_LOGIC_VECTOR (23 downto 0);
                           YYB : in  STD_LOGIC_VECTOR (23 downto 0);
                           XXB : out STD_LOGIC_VECTOR (23 downto 0)
                           );
           END COMPONENT;
--
signal W211BL,W212BL,W213BL,W221BL,W222BL,W223BL : std_logic_vector(23 downto 0);
signal B21BL,B22BL,B23BL  : std_logic_vector(23 downto 0);
signal W311BL,W312BL,W313BL,W321BL,W322BL,W323BL,W331BL,W332BL,W333BL : std_logic_vector(23 downto 0);
signal B31BL,B32BL,B33BL  : std_logic_vector(23 downto 0);
signal W411BL,W412BL,W421BL,W422BL,W431BL,W432BL : std_logic_vector(23 downto 0);
signal B41BL,B42BL : std_logic_vector(23 downto 0);
--
--signal W211AL,W212AL,W213AL,W221AL,W222AL,W223AL : std_logic_vector(23 downto 0);
--signal B21AL,B22AL,B23AL  : std_logic_vector(23 downto 0);
--signal W311AL,W312AL,W313AL,W321AL,W322AL,W323AL,W331AL,W332AL,W333AL : std_logic_vector(23 downto 0);
--signal B31AL,B32AL,B33AL  : std_logic_vector(23 downto 0);
--signal W411AL,W412AL,W421AL,W422AL,W431AL,W432AL : std_logic_vector(23 downto 0);
--signal B41AL,B42AL : std_logic_vector(23 downto 0);
--
signal W211,W212,W213,W221,W222,W223,B21,B22,B23 : std_logic_vector(23 downto 0);
signal W311,W312,W313,W321,W322,W323,W331,W332,W333,B31,B32,B33 : std_logic_vector(23 downto 0);
signal W411,W412,W421,W422,W431,W432,B41,B42 : std_logic_vector(23 downto 0);
--
signal K1,K2,K11B,K12B,K13B,K21B,K22B,K23B : std_logic_vector(23 downto 0);
signal w211o, w212o, w213o, w221o, w222o, w223o : std_logic_vector(23 downto 0);
signal w311o, w312o, w313o, w321o, w322o, w323o, w331o, w332o, w333o : std_logic_vector(23 downto 0);
signal w411o, w412o, w421o, w422o, w431o, w432o : std_logic_vector(23 downto 0);
signal D21, D22, D23, D31, D32, D33, D41, D42 : std_logic_vector(23 downto 0);
```

```
signal DW211, DW212, DW213, DW221, DW222, DW223 : std_logic_vector(23 downto 0);
signal DW311, DW312, DW313, DW321, DW322, DW323, DW331, DW332, DW333 : std_logic_vector(23 downto 0);
signal DW411, DW412, DW421, DW422, DW431, DW432 : std_logic_vector(23 downto 0);
signal z21, z22, z23, z31, z32, z33, z41, z42 : std_logic_vector(23 downto 0);
signal a21, a22, a23, a31, a32, a33, a41int, a42int : std_logic_vector(23 downto 0);
signal a21B,a22B,a23B,a31B,a32B,a33B,a41B,a42B : std_logic_vector(23 downto 0);
signal A211B,A212B,A213B,A221B,A222B,A223B,A231B,A232B,A233B : std_logic_vector(23 downto 0);
signal A311B,A312B,A321B,A322B,A331B,A332B : std_logic_vector(23 downto 0);
signal T1, T2 : std_logic_vector(23 downto 0);
--
--signal K1F, K2F, a41f, a42f : real;
signal intcount2 : std_logic_vector(1 downto 0);

begin

--K1F <= real(CONV_INTEGER(signed(K1))) / 65536.0;
--K2F <= real(CONV_INTEGER(signed(K2))) / 65536.0;
--a41f <= real(CONV_INTEGER(signed(a41int))) / 65536.0;
--a42f <= real(CONV_INTEGER(signed(a42int))) / 65536.0;

COUNT2 <= intcount2;
-- 2BIT COUNTER
process (SYSCLK, RESET) begin
  if (RESET = '0') then intcount2 <= "00";
  elsif (SYSCLK'event and SYSCLK = '1') then
    intcount2 <= intcount2 + '1';
  end if;
end process;
-- K1, K2, T1, T2 GENERATOR
process(intcount2) begin
  case intcount2 is
    when "00"   => K1 <= "000010000000000000000000";  -- 8
                   K2 <= "000010000000000000000000";  -- 8
                                          T1 <= "000000010000000000000000";  -- 1
                                            T2 <= "000000000000000000000000";  -- 0
    when "01"   => K1 <= "000010000000000000000000";  -- 8
                   K2 <= "000001010000000000000000";  -- 5
                                            T1 <= "000000000000000000000000";  -- 0
                                            T2 <= "000000010000000000000000";  -- 1
    when "10"   => K1 <= "000001010000000000000000";  -- 5
                   K2 <= "000010000000000000000000";  -- 8
                                            T1 <= "000000000000000000000000";  -- 0
                                            T2 <= "000000010000000000000000";  -- 1
```

```
   when others => K1 <= "000001010000000000000000";  -- 5
                  K2 <= "000001010000000000000000";  -- 5
                                          T1 <= "000000000000000000000000";  -- 0
                                          T2 <= "000000010000000000000000";  -- 1

  end case;
end process;
-- NEURAL NET WITH BACK PROPAGATION
mulw211 : MUL24 PORT MAP (W211, K1, w211o, D21, DW211, K11B);
mulw212 : MUL24 PORT MAP (W212, K1, w212o, D22, DW212, K12B);
mulw213 : MUL24 PORT MAP (W213, K1, w213o, D23, DW213, K13B);
mulw221 : MUL24 PORT MAP (W221, K2, w221o, D21, DW221, K21B);
mulw222 : MUL24 PORT MAP (W222, K2, w222o, D22, DW222, K22B);
mulw223 : MUL24 PORT MAP (W223, K2, w223o, D23, DW223, K23B);
z21 <= signed(B21) + signed(w211o) + signed(w221o);
z22 <= signed(B22) + signed(w212o) + signed(w222o);
z23 <= signed(B23) + signed(w213o) + signed(w223o);
reluz21 : RELU_FUNC PORT MAP (z21, a21, a21B, D21);
reluz22 : RELU_FUNC PORT MAP (z22, a22, a22B, D22);
reluz23 : RELU_FUNC PORT MAP (z23, a23, a23B, D23);
-- W311 NO XXB atarino kasann kara tsuduki
a21B <= signed(A211B) + signed(A212B) + signed(A213B);
a22B <= signed(A221B) + signed(A222B) + signed(A223B);
a23B <= signed(A231B) + signed(A232B) + signed(A233B);
mulw311 : MUL24 PORT MAP (W311, a21, w311o, D31, DW311, A211B);
mulw312 : MUL24 PORT MAP (W312, a21, w312o, D32, DW312, A212B);
mulw313 : MUL24 PORT MAP (W313, a21, w313o, D33, DW313, A213B);
mulw321 : MUL24 PORT MAP (W321, a22, w321o, D31, DW321, A221B);
mulw322 : MUL24 PORT MAP (W322, a22, w322o, D32, DW322, A222B);
mulw323 : MUL24 PORT MAP (W323, a22, w323o, D33, DW323, A223B);
mulw331 : MUL24 PORT MAP (W331, a23, w331o, D31, DW331, A231B);
mulw332 : MUL24 PORT MAP (W332, a23, w332o, D32, DW332, A232B);
mulw333 : MUL24 PORT MAP (W333, a23, w333o, D33, DW333, A233B);
z31 <= signed(B31) + signed(w311o) + signed(w321o) + signed(w331o);
z32 <= signed(B32) + signed(w312o) + signed(w322o) + signed(w332o);
z33 <= signed(B33) + signed(w313o) + signed(w323o) + signed(w333o);
reluz31 : RELU_FUNC PORT MAP (z31, a31, a31B, D31);
reluz32 : RELU_FUNC PORT MAP (z32, a32, a32B, D32);
reluz33 : RELU_FUNC PORT MAP (z33, a33, a33B, D33);
a31B <= signed(A311B) + signed(A312B);
a32B <= signed(A321B) + signed(A322B);
a33B <= signed(A331B) + signed(A332B);
mulw411 : MUL24 PORT MAP (W411, a31, w411o, D41, DW411, A311B);
```

```
mulw412 : MUL24 PORT MAP (W412, a31, w412o, D42, DW412, A312B);
mulw421 : MUL24 PORT MAP (W421, a32, w421o, D41, DW421, A321B);
mulw422 : MUL24 PORT MAP (W422, a32, w422o, D42, DW422, A322B);
mulw431 : MUL24 PORT MAP (W431, a33, w431o, D41, DW431, A331B);
mulw432 : MUL24 PORT MAP (W432, a33, w432o, D42, DW432, A332B);
z41 <= signed(B41) + signed(w411o) + signed(w421o) + signed(w431o);
z42 <= signed(B42) + signed(w412o) + signed(w422o) + signed(w432o);
reluz41 : RELU_FUNC PORT MAP (z41, a41int, a41B, D41);
reluz42 : RELU_FUNC PORT MAP (z42, a42int, a42B, D42);
-- ERROR FEEDBACK
a41B <= signed(a41int) - signed(T1);
a42B <= signed(a42int) - signed(T2);
-- OUTPUT SEISEI
A41 <= a41int;
A42 <= a42int;
-- WEIGHT UPDATER
process(SYSCLK, RESET)
variable F211, F212, F213, F221, F222, F223 : std_logic_vector(9 downto 0);
variable F311, F312, F313, F321, F322, F323, F331, F332, F333 : std_logic_vector(9 downto 0);
variable F411, F412, F421, F422, F431, F432 : std_logic_vector(9 downto 0);
variable F21, F22, F23, F31, F32, F33, F41, F42 : std_logic_vector(9 downto 0);
begin
  if (RESET = '0') then
    W211 <= W211BL; W212 <= W212BL; W213 <= W213BL;
          W221 <= W221BL; W222 <= W222BL; W223 <= W223BL;
    B21  <= B21BL;  B22  <= B22BL;  B23  <= B23BL;
    W311 <= W311BL; W312 <= W312BL; W313 <= W313BL;
    W321 <= W321BL; W322 <= W322BL; W323 <= W323BL;
    W331 <= W331BL; W332 <= W332BL; W333 <= W333BL;
    B31  <= B31BL;  B32  <= B32BL;  B33  <= B33BL;
    W411 <= W411BL; W412 <= W412BL; W421 <= W421BL;
    W422 <= W422BL; W431 <= W431BL; W432 <= W432BL;
    B41  <= B41BL;  B42  <= B42BL;
  elsif (SYSCLK'event and SYSCLK = '1') then
    F211 := (others => DW211(23)); W211 <= signed(W211) - signed(F211 & DW211(23 downto 10));
          F212 := (others => DW212(23)); W212 <= signed(W212) - signed(F212 & DW212(23 downto 10));
          F213 := (others => DW213(23)); W213 <= signed(W213) - signed(F213 & DW213(23 downto 10));
          F221 := (others => DW221(23)); W221 <= signed(W221) - signed(F221 & DW221(23 downto 10));
    F222 := (others => DW222(23)); W222 <= signed(W222) - signed(F222 & DW222(23 downto 10));
          F223 := (others => DW223(23)); W223 <= signed(W223) - signed(F223 & DW223(23 downto 10));
    F21  := (others => D21(23));   B21  <= signed(B21)  - signed(F21  & D21(23 downto 10));
          F22  := (others => D22(23));   B22  <= signed(B22)  - signed(F22  & D22(23 downto 10));
          F23  := (others => D23(23));   B23  <= signed(B23)  - signed(F23  & D23(23 downto 10));
```

```
          F311 := (others => DW311(23)); W311 <= signed(W311) - signed(F311 & DW311(23 downto 10));
          F312 := (others => DW312(23)); W312 <= signed(W312) - signed(F312 & DW312(23 downto 10));
          F313 := (others => DW313(23)); W313 <= signed(W313) - signed(F313 & DW313(23 downto 10));
          F321 := (others => DW321(23)); W321 <= signed(W321) - signed(F321 & DW321(23 downto 10));
          F322 := (others => DW322(23)); W322 <= signed(W322) - signed(F322 & DW322(23 downto 10));
          F323 := (others => DW323(23)); W323 <= signed(W323) - signed(F323 & DW323(23 downto 10));
          F331 := (others => DW331(23)); W331 <= signed(W331) - signed(F331 & DW331(23 downto 10));
          F332 := (others => DW332(23)); W332 <= signed(W332) - signed(F332 & DW332(23 downto 10));
          F333 := (others => DW333(23)); W333 <= signed(W333) - signed(F333 & DW333(23 downto 10));
          F31  := (others => D31(23));   B31  <= signed(B31)  - signed(F31  & D31(23 downto 10));
          F32  := (others => D32(23));   B32  <= signed(B32)  - signed(F32  & D32(23 downto 10));
          F33  := (others => D33(23));   B33  <= signed(B33)  - signed(F33  & D33(23 downto 10));
          F411 := (others => DW411(23)); W411 <= signed(W411) - signed(F411 & DW411(23 downto 10));
          F412 := (others => DW412(23)); W412 <= signed(W412) - signed(F412 & DW412(23 downto 10));
          F421 := (others => DW421(23)); W421 <= signed(W421) - signed(F421 & DW421(23 downto 10));
          F422 := (others => DW422(23)); W422 <= signed(W422) - signed(F422 & DW422(23 downto 10));
          F431 := (others => DW431(23)); W431 <= signed(W431) - signed(F431 & DW431(23 downto 10));
          F432 := (others => DW432(23)); W432 <= signed(W432) - signed(F432 & DW432(23 downto 10));
  end if;
end process;
-- CONSTANS
-- BEFORE
W211BL <= CONV_std_logic_vector(integer( 0.1*65536.0),24);
W212BL <= CONV_std_logic_vector(integer( 0.3*65536.0),24);
W213BL <= CONV_std_logic_vector(integer( 0.6*65536.0),24);
W221BL <= CONV_std_logic_vector(integer( 0.4*65536.0),24);
W222BL <= CONV_std_logic_vector(integer( 0.5*65536.0),24);
W223BL <= CONV_std_logic_vector(integer( 0.1*65536.0),24);
B21BL  <= CONV_std_logic_vector(integer(-1.0*65536.0),24);
B22BL  <= CONV_std_logic_vector(integer(-1.0*65536.0),24);
B23BL  <= CONV_std_logic_vector(integer(-1.0*65536.0),24);
W311BL <= CONV_std_logic_vector(integer( 0.1*65536.0),24);
W312BL <= CONV_std_logic_vector(integer( 0.3*65536.0),24);
W313BL <= CONV_std_logic_vector(integer( 0.6*65536.0),24);
W321BL <= CONV_std_logic_vector(integer( 0.4*65536.0),24);
W322BL <= CONV_std_logic_vector(integer( 0.5*65536.0),24);
W323BL <= CONV_std_logic_vector(integer( 0.1*65536.0),24);
W331BL <= CONV_std_logic_vector(integer( 0.4*65536.0),24);
W332BL <= CONV_std_logic_vector(integer( 0.5*65536.0),24);
W333BL <= CONV_std_logic_vector(integer( 0.1*65536.0),24);
B31BL  <= CONV_std_logic_vector(integer(-1.0*65536.0),24);
B32BL  <= CONV_std_logic_vector(integer(-1.0*65536.0),24);
B33BL  <= CONV_std_logic_vector(integer(-1.0*65536.0),24);
```

```
W411BL <= CONV_std_logic_vector(integer( 0.7*65536.0),24);
W412BL <= CONV_std_logic_vector(integer( 0.2*65536.0),24);
W421BL <= CONV_std_logic_vector(integer( 0.2*65536.0),24);
W422BL <= CONV_std_logic_vector(integer( 0.5*65536.0),24);
W431BL <= CONV_std_logic_vector(integer( 1.3*65536.0),24);
W432BL <= CONV_std_logic_vector(integer( 1.1*65536.0),24);
B41BL  <= CONV_std_logic_vector(integer(-1.0*65536.0),24);
B42BL  <= CONV_std_logic_vector(integer(-1.0*65536.0),24);
-- AFTER
--W211AL <= CONV_std_logic_vector(integer( 0.1666*65536.0),24);
--W212AL <= CONV_std_logic_vector(integer(-0.6927*65536.0),24);
--W213AL <= CONV_std_logic_vector(integer(-0.9238*65536.0),24);
--W221AL <= CONV_std_logic_vector(integer( 0.5176*65536.0),24);
--W222AL <= CONV_std_logic_vector(integer( 0.7890*65536.0),24);
--W223AL <= CONV_std_logic_vector(integer( 0.7730*65536.0),24);
--B21AL  <= CONV_std_logic_vector(integer(-0.6254*65536.0),24);
--B22AL  <= CONV_std_logic_vector(integer(-1.4991*65536.0),24);
--B23AL  <= CONV_std_logic_vector(integer(-1.4630*65536.0),24);
--W311AL <= CONV_std_logic_vector(integer(-0.0478*65536.0),24);
--W312AL <= CONV_std_logic_vector(integer( 0.5434*65536.0),24);
--W313AL <= CONV_std_logic_vector(integer( 0.2638*65536.0),24);
--W321AL <= CONV_std_logic_vector(integer( 0.1203*65536.0),24);
--W322AL <= CONV_std_logic_vector(integer(-0.4388*65536.0),24);
--W323AL <= CONV_std_logic_vector(integer(-0.0178*65536.0),24);
--W331AL <= CONV_std_logic_vector(integer( 0.3223*65536.0),24);
--W332AL <= CONV_std_logic_vector(integer( 0.8529*65536.0),24);
--W333AL <= CONV_std_logic_vector(integer( 0.0157*65536.0),24);
--B31AL  <= CONV_std_logic_vector(integer(-1.7412*65536.0),24);
--B32AL  <= CONV_std_logic_vector(integer(-1.4000*65536.0),24);
--B33AL  <= CONV_std_logic_vector(integer(-0.5186*65536.0),24);
--W411AL <= CONV_std_logic_vector(integer( 0.6628*65536.0),24);
--W412AL <= CONV_std_logic_vector(integer(-0.3784*65536.0),24);
--W421AL <= CONV_std_logic_vector(integer( 0.9149*65536.0),24);
--W422AL <= CONV_std_logic_vector(integer(-1.3305*65536.0),24);
--W431AL <= CONV_std_logic_vector(integer( 1.7372*65536.0),24);
--W432AL <= CONV_std_logic_vector(integer( 0.4477*65536.0),24);
--B41AL  <= CONV_std_logic_vector(integer(-1.4556*65536.0),24);
--B42AL  <= CONV_std_logic_vector(integer( 1.3607*65536.0),24);
end Behavioral;
```