library IEEE;

use IEEE.STD\_LOGIC\_1164.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 PROJECT is

Port ( A : in STD\_LOGIC\_VECTOR (3 downto 0);

B : in STD\_LOGIC\_VECTOR (3 downto 0);

S : in STD\_LOGIC\_VECTOR (3 downto 0);

X : inout STD\_LOGIC\_VECTOR (3 downto 0);

Z : out STD\_LOGIC;

OVERFLOW : out STD\_LOGIC;

C : inout STD\_LOGIC\_VECTOR (5 downto 0);

F : inout STD\_LOGIC\_VECTOR (7 downto 0));

end PROJECT;

architecture Behavioral of PROJECT is

begin

PROCESS (S)

begin

IF(S="0000") THEN

F(7) <='0';

F(6) <='0';

F(5) <='0';

F(4) <='0';

F(3) <='0';

F(2) <='0';

F(1) <='0';

F(0) <='0';

ELSIF(S="0001") THEN

F(7) <='0';

F(6) <='0';

F(5) <='0';

F(4) <='0';

F(3) <=(A(3) NOR B(3));

F(2) <=(A(2) NOR B(2));

F(1) <=(A(1) NOR B(1));

F(0) <=(A(0) NOR B(0));

ELSIF(S="0010") THEN

C(0) <=(A(0) AND B(1)) AND (A(1) AND B(0));

C(1) <=((A(2) AND B(0)) AND (A(1) AND B(1))) OR ((A(2) AND B(0)) AND (A(0) AND B(2))) OR ((A(1) AND B(1)) AND (A(0) AND B(2))) OR ((A(2) AND B(0)) AND C(0)) OR ((A(1) AND B(1)) AND C(0)) OR ((A(0) AND B(2)) AND C(0));

C(2) <=((A(3) AND B(0)) AND (A(2) AND B(1))) OR ((A(3) AND B(0)) AND (A(1) AND B(2))) OR ((A(3) AND B(0)) AND (A(0) AND B(3))) OR ((A(3) AND B(0)) AND C(1)) OR ((A(2) AND B(1)) AND C(1)) OR ((A(1) AND B(2)) AND C(1)) OR ((A(0) AND B(3)) AND C(1)) OR ((A(2) AND B(1)) AND (A(1) AND B(2))) OR ((A(2) AND B(1)) AND (A(0) AND B(3))) OR ((A(1) AND B(2)) AND (A(0) AND B(3)));

C(3) <=((A(3) AND B(1)) AND (A(2) AND B(2))) OR ((A(3) AND B(1)) AND (A(1) AND B(3))) OR ((A(2) AND B(2)) AND (A(1) AND B(3))) OR ((A(3) AND B(1)) AND C(2)) OR ((A(1) AND B(3)) AND C(2)) OR ((A(2) AND B(2)) AND C(2));

C(4) <=((A(3) AND B(2)) AND (A(2) AND B(3))) OR ((A(3) AND B(2)) AND C(3)) OR ((A(2) AND B(3)) AND C(3));

C(5) <=((A(3) AND B(3)) AND C(4));

F(0) <=A(0) AND B(0);

F(1) <=(A(1) AND B(0)) XOR (A(0) AND B(1));

F(2) <=(A(2) AND B(0)) XOR (A(1) AND B(1)) XOR (A(0) AND B(2)) XOR C(0);

F(3) <=(A(3) AND B(0)) XOR (A(2) AND B(1)) XOR (A(1) AND B(2)) XOR (A(0) AND B(3)) XOR C(1);

F(4) <=(A(3) AND B(1)) XOR (A(2) AND B(2)) XOR (A(1) AND B(3)) XOR C(2);

F(5) <=(A(3) AND B(2)) XOR (A(2) AND B(3)) XOR C(3);

F(6) <=(A(3) AND B(3)) XOR C(4);

F(7) <=C(5);

ELSIF(S="0011") THEN

F(7) <='0';

F(6) <='0';

F(5) <='0';

F(4) <='0';

F(3) <=(NOT B(3));

F(2) <=(NOT B(2));

F(1) <=(NOT B(1));

F(0) <=(NOT B(0));

ELSIF(S="0100") THEN

F(0) <=A(3);

F(1) <=A(2);

F(2) <=A(1);

F(3) <=A(0);

F(4) <='0';

F(5) <='0';

F(6) <='0';

F(7) <='0';

ELSIF(S="0101") THEN

F(7) <='0';

F(6) <='0';

F(5) <='0';

F(4) <='0';

F(3) <=(NOT A(3));

F(2) <=(NOT A(2));

F(1) <=(NOT A(1));

F(0) <=(NOT A(0));

ELSIF(S="0110") THEN

F(7) <='0';

F(6) <='0';

F(5) <='0';

F(4) <='0';

F(3) <=(A(3) XOR B(3));

F(2) <=(A(2) XOR B(2));

F(1) <=(A(1) XOR B(1));

F(0) <=(A(0) XOR B(0));

ELSIF(S="0111") THEN

F(7) <='0';

F(6) <='0';

F(5) <='0';

F(4) <='0';

F(3) <=(A(3) NAND B(3));

F(2) <=(A(2) NAND B(2));

F(1) <=(A(1) NAND B(1));

F(0) <=(A(0) NAND B(0));

ELSIF(S="1000") THEN

F(7) <='0';

F(6) <='0';

F(5) <='0';

F(4) <='0';

F(3) <=(A(3) AND B(3));

F(2) <=(A(2) AND B(2));

F(1) <=(A(1) AND B(1));

F(0) <=(A(0) AND B(0));

ELSIF(S="1001") THEN

F(7) <='0';

F(6) <='0';

F(5) <='0';

F(4) <='0';

F(3) <=(A(3) XNOR B(3));

F(2) <=(A(2) XNOR B(2));

F(1) <=(A(1) XNOR B(1));

F(0) <=(A(0) XNOR B(0));

ELSIF(S="1010") THEN

F(7) <='0';

F(6) <='0';

F(5) <='0';

F(4) <=A(3);

F(3) <=A(2);

F(2) <=A(1);

F(1) <=A(0);

F(0) <='0';

ELSIF(S="1011") THEN

X(0) <=B(0) XOR '0';

X(1) <=B(1) XOR '0';

X(2) <=B(2) XOR '0';

X(3) <=B(3) XOR '0';

C(0) <= '0';

C(1) <=(A(0) AND X(0)) OR (A(0) AND C(0)) OR (X(0) AND C(0));

C(2) <=(A(1) AND X(1)) OR (A(1) AND C(1)) OR (X(1) AND C(1));

C(3) <=(A(2) AND X(2)) OR (A(2) AND C(2)) OR (X(2) AND C(2));

C(4) <=(A(3) AND X(3)) OR (A(3) AND C(3)) OR (X(3) AND C(3));

C(5) <='0';

F(0) <=A(0) XOR X(0) XOR C(0);

F(1) <=A(1) XOR X(1) XOR C(1);

F(2) <=A(2) XOR X(2) XOR C(2);

F(3) <=A(3) XOR X(3) XOR C(3);

F(4) <=C(4);

F(5) <='0';

F(6) <='0';

F(7) <='0';

OVERFLOW <=(A(3) AND X(3) AND (NOT F(3))) OR ((NOT A(3)) AND (NOT X(3)) AND F(3));

IF(F="00000000") THEN

Z <='1';

ELSE

Z <='0';

END IF;

ELSIF(S="1100") THEN

F(7) <='0';

F(6) <='0';

F(5) <='0';

F(4) <=B(3);

F(3) <=B(2);

F(2) <=B(1);

F(1) <=B(0);

F(0) <='0';

ELSIF(S="1101") THEN

X(0) <=B(0) XOR '1';

X(1) <=B(1) XOR '1';

X(2) <=B(2) XOR '1';

X(3) <=B(3) XOR '1';

C(0) <= '1';

C(1) <=(A(0) AND X(0)) OR (A(0) AND C(0)) OR (X(0) AND C(0));

C(2) <=(A(1) AND X(1)) OR (A(1) AND C(1)) OR (X(1) AND C(1));

C(3) <=(A(2) AND X(2)) OR (A(2) AND C(2)) OR (X(2) AND C(2));

C(4) <=(A(3) AND X(3)) OR (A(3) AND C(3)) OR (X(3) AND C(3));

C(5) <='0';

F(0) <=A(0) XOR X(0) XOR C(0);

F(1) <=A(1) XOR X(1) XOR C(1);

F(2) <=A(2) XOR X(2) XOR C(2);

F(3) <=A(3) XOR X(3) XOR C(3);

F(4) <=NOT C(4);

F(5) <='0';

F(6) <='0';

F(7) <='0';

OVERFLOW <=(A(3) AND X(3) AND (NOT F(3))) OR ((NOT A(3)) AND (NOT X(3)) AND F(3));

IF(F="00000000") THEN

Z <='1';

ELSE

Z <='0';

END IF;

ELSIF(S="1110") THEN

F(7) <='0';

F(6) <='0';

F(5) <='0';

F(4) <='0';

F(3) <=(A(3) OR B(3));

F(2) <=(A(2) OR B(2));

F(1) <=(A(1) OR B(1));

F(0) <=(A(0) OR B(0));

ELSIF(S="1111") THEN

F(7) <='1';

F(6) <='1';

F(5) <='1';

F(4) <='1';

F(3) <='1';

F(2) <='1';

F(1) <='1';

F(0) <='1';

END IF;

END PROCESS;

end Behavioral;