**fulladd**

LIBRARY ieee ;

USE ieee.std\_logic\_1164.all ;

ENTITY fulladd IS

PORT ( Cin, x, y : IN STD\_LOGIC ;

s, Cout : OUT STD\_LOGIC ) ;

END fulladd ;

ARCHITECTURE LogicFunc OF fulladd IS

BEGIN

s <= x XOR y XOR Cin ;

Cout <= (x AND y) OR (Cin AND x) OR (Cin AND y) ;

END LogicFunc ;

**fulladd\_package**

LIBRARY ieee ;

USE ieee.std\_logic\_1164.all ;

PACKAGE fulladd\_package IS

COMPONENT fulladd

PORT ( Cin, x, y : IN STD\_LOGIC ;

s, Cout : OUT STD\_LOGIC ) ;

END COMPONENT ;

END fulladd\_package ;

**seven\_segment\_display**

Library ieee;

USE ieee.std\_logic\_1164.all;

ENTITY seven\_segment\_display IS

PORT (W,X,Y,Z : IN STD\_LOGIC;

A,B,C,D,E,F,G : OUT STD\_LOGIC);

END seven\_segment\_display;

ARCHITECTURE LogicFunc OF seven\_segment\_display IS

BEGIN

A <= (X AND NOT Y AND NOT Z) OR (W AND X AND NOT Y) OR (NOT W AND NOT X AND NOT Y AND Z) OR (W AND NOT X AND Y AND Z);

B <= (X AND Y AND NOT Z) OR (W AND X AND Y) OR (W AND Y AND Z) OR (W AND X AND NOT Z) OR (NOT W AND X AND NOT Y AND Z);

C <= (NOT W AND NOT X AND Y AND NOT Z) OR (W AND X AND NOT Z) OR (W AND X AND Y);

D <= (NOT X AND NOT Y AND Z) OR (NOT W AND X AND NOT Y AND NOT Z) OR (X AND Y AND Z) OR (W AND NOT X AND Y AND NOT Z);

E <= (NOT W AND Z) OR (NOT W AND X AND NOT Y) OR (NOT X AND NOT Y AND Z);

F <= (NOT W AND NOT X AND Z) OR (NOT W AND NOT X AND Y) OR (NOT W AND Y AND Z) OR (W AND X AND NOT Y);

G <= (NOT W AND NOT X AND NOT Y) OR (NOT W AND X AND Y AND Z);

END LogicFunc;

**seven\_segment\_display\_package**

LIBRARY ieee ;

USE ieee.std\_logic\_1164.all ;

PACKAGE seven\_segment\_display\_package IS

COMPONENT fulladd

PORT ( W,X,Y,Z : IN STD\_LOGIC;

A,B,C,D,E,F,G : OUT STD\_LOGIC);

END COMPONENT ;

END seven\_segment\_display\_package ;

**Adder8**

LIBRARY ieee ;

USE ieee.std\_logic\_1164.all ;

USE work.fulladd\_package.all ;

USE work.seven\_segment\_display\_package.all ;

USE ieee.std\_logic\_signed.all ;

ENTITY adder8 IS

PORT ( Cin : IN STD\_LOGIC ;

X, Y : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0) ;

S : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0) ;

Cout : OUT STD\_LOGIC ;

WW,XX,YY,ZZ: IN STD\_LOGIC\_VECTOR(1 DOWNTO 0) ;

A,B,C,D,E,F,G : OUT STD\_LOGIC\_VECTOR(1 DOWNTO 0)) ;

END adder8 ;

ARCHITECTURE Structure OF adder8 IS

SIGNAL C : STD\_LOGIC\_VECTOR(1 TO 7) ;

BEGIN

stage0: fulladd PORT MAP ( Cin, X(0), Y(0), S(0), C(1) ) ;

stage1: fulladd PORT MAP ( C(1), X(1), Y(1), S(1), C(2) ) ;

stage2: fulladd PORT MAP ( C(2), X(2), Y(2), S(2), C(3) ) ;

stage3: fulladd PORT MAP ( C(3), X(3), Y(3), S(3), C(4) ) ;

stage4: fulladd PORT MAP ( C(4), X(4), Y(4), S(4), C(5) ) ;

stage5: fulladd PORT MAP ( C(5), X(5), Y(5), S(5), C(6) ) ;

stage6: fulladd PORT MAP ( C(6), X(6), Y(6), S(6), C(7) ) ;

stage7: fulladd PORT MAP ( C(7), X(7), Y(7), S(7), Cout ) ;

stage8:seven\_segment\_display PORT MAP (WW(1),XX(1),YY(1),ZZ(1),A(1),B(1),C(1),D(1),E(1),F(1),G(1));

stage9:seven\_segment\_display PORT MAP (WW(0),XX(0),YY(0),ZZ(0),A(0),B(0),C(0),D(0),E(0),F(0),G(0));

END Structure ;