LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY add IS

PORT(

a, b, c\_in : IN STD\_LOGIC;

c\_out, sum : OUT STD\_LOGIC);

END add;

ARCHITECTURE adder OF add IS

BEGIN

c\_out <=((a xor b) and c\_in) or (a and b);

sum <= (a xor b) xor c\_in;

END adder;

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY full\_add IS

PORT(

c0: IN STD\_LOGIC;

a, b: IN STD\_LOGIC\_VECTOR(4 downto 1);

c4: OUT STD\_LOGIC;

sum: OUT STD\_LOGIC\_VECTOR(4 downto 1));

END full\_add;

ARCHITECTURE adder OF full\_add IS

-- Component declaration

COMPONENT add

PORT( a, b, c\_in: IN STD\_LOGIC;

c\_out, sum: OUT STD\_LOGIC);

END COMPONENT;

-- Define a signal for internal carry bits

SIGNAL c : STD\_LOGIC\_VECTOR (4 downto 0);

BEGIN

c(0) <= c0;

adders:

FOR i IN 1 to 4 GENERATE

adder: add PORT MAP (a(i), b(i), c(i-1), c(i), sum(i));

END GENERATE;

c4 <= c(4);

END adder;