程式碼

library ieee;

use ieee.std\_logic\_1164.all;

use work.one\_bit\_ALU\_package.all;

use work.lab1\_2\_package.all;

entity lab3 is

port( ALUctrl : in std\_logic\_vector(3 downto 0);

A, B: in std\_logic\_vector(6 downto 0);

out1, out2, out3, out4, out5, out6, out7, out8, out9, out10, out11, out12, out13, out14 :out std\_logic);

end lab3;

Architecture logicfunc of lab3 is

signal set, ignore, carryin: std\_logic;

signal result, carryout: std\_logic\_vector(6 downto 0);

signal sign : std\_logic;

begin

sign <= ALUctrl(2) AND ALUctrl(1);

G1: For i in 0 to 6 generate

G2: if i=0 generate

stage1: one\_bit\_ALU port map(ALUctrl(3 downto 0), A(i), B(i), set, sign, result(i), ignore, carryout(i));

end generate;

G3: if i=1 or i=2 or i=3 or i=4 or i=5 generate

stage2: one\_bit\_ALU port map(ALUctrl(3 downto 0), A(i), B(i), '0', carryout(i-1), result(i), ignore, carryout(i));

end generate;

G4: if i=6 generate

stage3: one\_bit\_ALU port map(ALUctrl(3 downto 0), A(i), B(i), '0', carryout(i-1), result(i), set, carryout(i));

end generate;

end generate;

output: lab1\_2 port map('0', result(6), result(5), result(4), result(3), result(2), result(1), result(0), out8, out9, out10, out11, out12, out13, out14, out1, out2, out3, out4, out5, out6, out7);

end logicfunc;

心得

透過這次實驗，讓我更加了解VHDL語法以及ALU加法器的原理，雖然這次實驗一開始聽老師講解的時候完全不知道要幹嘛，只知道要做7-bit的ALU加法器，我甚至都不知道那是什麼，後來我們上網找資料，才大概知道ALU加法器在幹嘛，程式碼開頭下手很困難，中途程式碼檔案甚至消失了，不知為何資料夾裡有檔案，但整個專案內沒有，這次的實驗讓我更加熟悉這些操作