**微算機系統**

**小組專案報告**

實驗八

組別： 18

班級、姓名與學號：

資工二 蕭耕宏 110590005

資工二 楊榮鈞 110590034

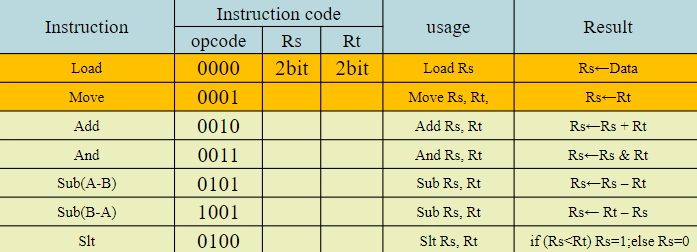
日期： 2022.12.20

1. 實驗內容：

本次實驗要實作一個簡易cpu，共有六個暫存器(R0,R1,R2,R3,RS,RT)，一個bus，需用HEX顯示RS、RT及BUS的值。

根據這次實做給的opcode，我們這次要有Load、Move、Add、And、Sub(A-B)、Sub(B-A)、Slt的功能。

1. 實驗過程及結果：

operations 

根據老師給的這張表寫出來的code。

實驗的結果

|  |  |
| --- | --- |
|  |  |
| rt = r0 (00), rs = r0 (00), opcode = 0000,  data = 00000000, r0 = 00000000,  r1 = 00000000, r2 = 00000000,  r3 = 00000000 | rt = r0 (00), rs = r1 (01), opcode = 0000,  data = 00000010, r0 = 00000000,  r1 = 00000010, r2 = 00000000,  r3 = 00000000 |
|  |  |
| rt = r1 (01), rs = r2 (10), opcode = 0000,  data = 00000011, r0 = 00000000,  r1 = 00000010, r2 = 00000011,  r3 = 00000000 | rt = r2 (10), rs = r3 (11), opcode = 0000,  data = 00000100, r0 = 00000000,  r1 = 00000010, r2 = 00000011,  r3 = 00000100 |
|  |  |
| rt = r1 (01), rs = r3 (11), opcode = 0010,  data = 00000100, r0 = 00000000,  r1 = 00000010, r2 = 00000011,  r3 = 00000110 | rt = r1 (01), rs = r3 (11), opcode = 0011,  data = 00000100, r0 = 00000000,  r1 = 00000010, r2 = 00000011,  r3 = 00000010 |
|  |  |
| rt = r1 (01), rs = r3 (11), opcode = 0101,  data = 00000100, r0 = 00000000,  r1 = 00000010, r2 = 00000011,  r3 = 00000000 | rt = r1 (01), rs = r3 (11), opcode = 1001,  data = 00000100, r0 = 00000000,  r1 = 00000010, r2 = 00000011,  r3 = 00000010 |
|  |  |
| rt = r1 (01), rs = r3 (11), opcode = 0100,  data = 00000100, r0 = 00000000,  r1 = 00000010, r2 = 00000011,  r3 = 00000000 | rt = r1 (01), rs = r3 (11), opcode = 0001,  data = 00000100, r0 = 00000000,  r1 = 00000010, r2 = 00000011,  r3 = 00000010 |
|  |  |
| rt = r2 (10), rs = r3 (11), opcode = 0100,  data = 00000100, r0 = 00000000,  r1 = 00000010, r2 = 00000011,  r3 = 00000001 | rt = r0 (00), rs = r3 (11), opcode = 0100,  data = 00000100, r0 = 00000000,  r1 = 00000010, r2 = 00000011,  r3 = 00000000 |

1. 程式碼

|  |
| --- |
| 進階題 |
| Lab8.vhd |
| library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  use ieee.numeric\_std.all;  use work.lab8\_package.all;  entity Lab8 is  port( clk : in std\_logic;  rs, rt : in std\_logic\_vector(1 downto 0);  op : in std\_logic\_vector(3 downto 0);  data : in std\_logic\_vector(7 downto 0);  -- hex0 ~ hex5  bus0, bus1, rs2, rs3, rt4, rt5 :out std\_logic\_vector(0 to 6)  );  end Lab8;  architecture behavior of Lab8 is  signal r0, r1, r2, r3 : std\_logic\_vector(7 downto 0);  shared variable t1, t2 : std\_logic\_vector(7 downto 0);  -- t1 -> rs, t2 -> rt  begin  process  begin  wait until clk'event and clk = '1';  if rs = "00" then  t1 := r0;  elsif rs = "01" then  t1 := r1;  elsif rs = "10" then  t1 := r2;  elsif rs = "11" then  t1 := r3;  end if;    if rt = "00" then  t2 := r0;  elsif rt = "01" then  t2 := r1;  elsif rt = "10" then  t2 := r2;  elsif rt = "11" then  t2 := r3;  end if;    if op = "0000" then  t1 := data;    elsif op = "0001" then  t1 := t2;    elsif op = "0010" then  t1 := t1 + t2;    elsif op = "0011" then  t1 := t1 and t2;    elsif op = "0101" then  t1 := t1 - t2;    elsif op = "1001" then  t1 := t2 - t1;    elsif op = "0100" then  t1 := t1 - t2;  if t1(7) = '1' then  t1 := "00000001";  else  t1 := "00000000";  end if;    end if;    if rs = "00" then  r0 <= t1;  elsif rs = "01" then  r1 <= t1;  elsif rs = "10" then  r2 <= t1;  elsif rs = "11" then  r3 <= t1;  end if;    end process;  stage0 : hex port map(data(7 downto 4), data(3 downto 0),  bus1(0 to 6), bus0(0 to 6));  stage1 : hex port map(t1(7 downto 4), t1(3 downto 0),  rs3(0 to 6), rs2(0 to 6));  stage2 : hex port map(t2(7 downto 4), t2(3 downto 0),  rt5(0 to 6), rt4(0 to 6));  end behavior; |
| hex.vhd |
| library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_signed.all;  entity hex is  port( sw1 : in std\_logic\_vector(3 downto 0);  sw0 : in std\_logic\_vector(3 downto 0);  hex1 :out std\_logic\_vector(0 to 6);  hex0 :out std\_logic\_vector(0 to 6)  );  end hex;  architecture behavior of hex is  begin  with sw1(3 downto 0) select  hex1 <= "0000001" when "0000",  "1001111" when "0001",  "0010010" when "0010",  "0000110" when "0011",  "1001100" when "0100",  "0100100" when "0101",  "0100000" when "0110",  "0001111" when "0111",  "0000000" when "1000",  "0001100" when "1001",  "0001000" when "1010",  "1100000" when "1011",  "1110010" when "1100",  "1000010" when "1101",  "0110000" when "1110",  "0111000" when "1111",  "1111111" when others;  with sw0(3 downto 0) select  hex0 <= "0000001" when "0000",  "1001111" when "0001",  "0010010" when "0010",  "0000110" when "0011",  "1001100" when "0100",  "0100100" when "0101",  "0100000" when "0110",  "0001111" when "0111",  "0000000" when "1000",  "0001100" when "1001",  "0001000" when "1010",  "1100000" when "1011",  "1110010" when "1100",  "1000010" when "1101",  "0110000" when "1110",  "0111000" when "1111",  "1111111" when others;  end behavior; |
| lab8\_package.vhd |
| library ieee;  use ieee.std\_logic\_1164.all;  package lab8\_package is  component hex  port( sw1 : in std\_logic\_vector(3 downto 0);  sw0 : in std\_logic\_vector(3 downto 0);  hex1: out std\_logic\_vector(0 to 6);  hex0: out std\_logic\_vector(0 to 6)  );  end component hex;  end lab8\_package; |

1. 本次實驗過程說明與解決方法:

實驗過程:

原本想使用port map來組成簡易cpu，但開始做的時候發現非常困難，所以之後就改用一個process來做。先寫完虛擬碼後再開始做實驗，我們做法是根據opcode判斷，rs和rt如何運作，用兩個變數接rs和rt指向的暫存器的數值，之後把存在rs的運算結果write回到rs指向的暫存器，最後把bus，rs和rt輸出到hex上。

這次遇到的困難是想要使用port map，但是很難用port map 進行即時運算。

解決方式：

我們選擇使用process還有shared variable進行即時運算，port map的部分用在七段顯示器的輸出來解決這次的問題。