4x4 鍵盤掃描輸入 4 位數 VHDL- keyin_4word.vhd (鍵盤輸入信號 com 點接高電位)

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
entity keyin_4word is
port(
        clr : in std_logic;    --位移暫存器電路之清除腳
        ena : in std_logic;      --位移暫存器電路之載入致能腳
        clk1: in std_logic;      --鍵盤掃描及彈跳處理取樣信號(40HZ~250HZ)
        y : in std_logic_vector(3 downto 0);    --Y  列輸入信號
        x : out std_logic_vector(3 downto 0);    --X  行掃描輸出信號
        scan_seg : out std_logic_vector(3 downto 0);      --Scan_seg  七節顯示器掃瞄輸出信號
        seg : out std_logic_vector(0 to 6)) ;
end keyin_4word;
--****************************************
architecture key of keyin_4word is
signal bcd        :       std_logic_vector(3 downto 0);
signal y_code : std_logic_vector(1 downto 0);
signal keycode : std_logic_vector(3 downto 0);
signal deb_press : std_logic;
signal sh_reg : std_logic_vector(15 downto 0);
--*****************************************
component debounce
PORT(
          fs:IN STD_LOGIC;
          bounce_in:in std_logic;
          deb_out:out STD_LOGIC
        );
end component;
--****************************
begin
key_code:      --按鍵偵測及編碼電路
process(clk1)
begin
  if clk1='1' and clk1'event then
        if y="1111" then
                if y_code="11" then
                        y_code<="00";
                else y_code<=y_code+1;
                end if;
        elsif y="1110" then
                keycode<=y_code&"00";
```

```
        elsif y="1101" then
                keycode<=y_code&"01";
        elsif y="1011" then
                keycode<=y_code&"10";
        elsif y="0111" then
                keycode<=y_code&"11";
        end if;
  end if;
end process key_code;
--*****************************
key_scan:      --列掃描信號輸出電路
process(y_code)
begin
        case y_code is
        when "00" => x<="0111";
        when "01" => x<="1011";
        when "10" => x<="1101";
        when "11" => x<="1110";
        when others=>x<="1111";
        end case;
end process key_scan;
--*****************************
load_4word:
block
signal press : std_logic;
begin
press<=(y(0) and y(1) and y(2) and y(3));    --按鍵未按下為 1 按下為 0
debounce_mod:debounce
        port map(fs=>clk1,bounce_in=>press,deb_out=>deb_press);
end block load_4word;
--*******************************************************
shift:process(clr,deb_press)
begin
        if clr='0' then      --當 CLR 為低電位時，暫存器被清除為 0
                sh_reg(15 downto 0)<="0000000000000000";
        elsif deb_press='0' and deb_press'event then      --按鍵防彈跳負緣信號觸發
          if ena='0' then    --且載入致能 ena 為低電位時
                sh_reg(15 downto 0)<= sh_reg(11 downto 0) & keycode;
                                --輸入個位數字(4bit)其餘位元向左移位 4bit
          end if;
        end if;
end process shift;
--*****************************
```

呼叫防彈跳程式 debounce

block 區塊，可將區塊描述及零散敘述整合成一個 block，增加程式的可讀性。

```
Scan_seg7:
process(Clk1)
  variable Scan1 : std_logic_vector(1 downto 0):="00";
    begin
        wait until clk1='1' ;
        if (Scan1="00") then
                Scan_seg <= "0111" ;
                bcd <= sh_reg(15 downto 12) ;
        elsif (Scan1="01") then
                Scan_seg <= "1011" ;
                bcd <= sh_reg(11 downto 8);
        elsif (Scan1="10") then
                Scan_seg <= "1101" ;
                bcd <= sh_reg(7 downto 4);
        else Scan_seg <= "1110" ;
                bcd <= sh_reg(3 downto 0);
        end if ;
        if Scan1 >= "11" then Scan1 := "00" ;
        else Scan1 := Scan1 + 1 ;
        end if ;
end process scan_seg7 ;
--*******************************************************
with bcd select
  seg <= "1111110"    when "0000" ,
             "0110000"    when "0001" ,
             "1101101"    when "0010" ,
             "1111001"    when "0011" ,
             "0110011"    when "0100" ,
             "1011011"    when "0101" ,
             "0011111"    when "0110" ,
             "1110000"    when "0111" ,
             "1111111"    when "1000" ,
             "1110011"    when "1001" ,
             "1001111"    when others ;
    end key    ;
```

4x4 鍵盤掃描輸入 4 位數 VHDL keyin_4word_com0.vhd (鍵盤輸入信號 com 點接低電位)

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
entity keyin_4word_com0 is
port(
	clr : in std_logic;
	ena : in std_logic;
	clk1: in std_logic;--鍵盤掃描及彈跳處理取樣信號(40HZ~250HZ)
	y : in std_logic_vector(3 downto 0);--Y 列輸入信號
	x : out std_logic_vector(3 downto 0);--X 行掃描輸出信號
	scan_seg : out std_logic_vector(3 downto 0);--Scan_seg 七節顯示器掃瞄輸出信號
	seg : out std_logic_vector(0 to 6)) ;
end keyin_4word_com0;
--****************************************
architecture key of keyin_4word_com0 is
signal bcd	:	std_logic_vector(3 downto 0);
signal y_code : std_logic_vector(1 downto 0);
signal keycode : std_logic_vector(3 downto 0);
signal deb_press : std_logic;
signal sh_reg : std_logic_vector(15 downto 0);
--*****************************************
component debounce
PORT(
	 fs:IN STD_LOGIC;
	 bounce_in:in std_logic;
	 deb_out:out STD_LOGIC
	);
end component;
--****************************
begin
key_code:	--按鍵偵測及編碼電路
process(clk1)
begin
  if clk1='1' and clk1'event then
	if y="0000" then
		if y_code="11" then
			y_code<="00";
		else y_code<=y_code+1;
		end if;
	elsif y="0001" then
		keycode<=y_code&"00";
```

```
        elsif y="0010" then
                keycode<=y_code&"01";
        elsif y="0100" then
                keycode<=y_code&"10";
        elsif y="1000" then
                keycode<=y_code&"11";
        end if;
  end if;
end process key_code;
--****************************
key_scan:--列掃描信號輸出電路
process(y_code)
begin
        case y_code is
        when "00" => x<="1000";
        when "01" => x<="0100";
        when "10" => x<="0010";
        when "11" => x<="0001";
        when others=>x<="0000";
        end case;
end process key_scan;

--****************************
load_4word:
block
signal press : std_logic;
begin
press<=(y(0) or y(1) or y(2) or y(3));      --按鍵未按下為 0 按下為 1
debounce_mod:debounce
        port map(fs=>clk1,bounce_in=>press,deb_out=>deb_press);
end block load_4word;
--*******************************************************
shift:process(clr,deb_press)
begin
        if clr='0' then
                sh_reg(15 downto 0)<="0000000000000000";
        elsif deb_press='1' and deb_press'event then      --按鍵防彈跳正緣信號觸發
          if ena='0' then
                sh_reg(15 downto 0)<= sh_reg(11 downto 0) & keycode;--輸入個位數字(4bit)其餘位元向左移位 4bit
          end if;
        end if;
end process shift;
--****************************
```

```
seg7:block    --把七節顯示器的掃描顯示電路組合成一個 block，增加程式可讀性
signal bcd      :      std_logic_vector(3 downto 0);    --將原為整體訊號之 bcd 改為區塊內訊號
begin
Scan_seg7:
process(Clk1)
  variable Scan1 : std_logic_vector(1 downto 0):="00";
   begin
      wait until clk1='1' ;
      if (Scan1="00") then
            Scan_seg <= "0111" ;
            bcd <= sh_reg(15 downto 12) ;
      elsif (Scan1="01") then
            Scan_seg <= "1011" ;
            bcd <= sh_reg(11 downto 8);
      elsif (Scan1="10") then
            Scan_seg <= "1101" ;
            bcd <= sh_reg(7 downto 4);
      else Scan_seg <= "1110" ;
            bcd <= sh_reg(3 downto 0);
      end if ;
      if Scan1 >= "11" then Scan1 := "00" ;
      else Scan1 := Scan1 + 1 ;
      end if ;
end process scan_seg7
--*********************************************************
with bcd select
  seg <= "1111110"    when "0000" ,
         "0110000"    when "0001" ,
         "1101101"    when "0010" ,
         "1111001"    when "0011" ,
         "0110011"    when "0100" ,
         "1011011"    when "0101" ,
         "0011111"    when "0110" ,
         "1110000"    when "0111" ,
         "1111111"    when "1000" ,
         "1110011"    when "1001" ,
         "1000000"    when "1010" ,
         "0100000"    when "1011" ,
         "0010000"    when "1100" ,
         "0001000"    when "1101" ,
         "0000100"    when "1110" ,
         "0000010"    when "1111" ,
         "1001111"    when others ;
end block seg7; --結束 7 節顯示器的顯示電路區塊
end key    ;
```

顯示 a,b,c,d,e 訊號

4x4 鍵盤掃描輸入 6 位數 VHDL- keyin_6word.vhd (鍵盤輸入信號 com 點接低電位)

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
entity keyin_6word is
port(
        clr : in std_logic;
        ena : in std_logic;
        clk1: in std_logic;    --鍵盤掃描及彈跳處理取樣信號(40HZ~250HZ)
        y : in std_logic_vector(3 downto 0);      --Y  列輸入信號
        x : out std_logic_vector(3 downto 0);      --X  行掃描輸出信號
        scan_seg : out std_logic_vector(5 downto 0);      --Scan_seg  七節顯示器掃瞄輸出信號
        seg : out std_logic_vector(0 to 6)) ;    --Scan_seg  七節顯示器輸出信號
end keyin_6word;
--****************************************
architecture key of keyin_6word is
signal bcd        :       std_logic_vector(3 downto 0);
signal y_code : std_logic_vector(1 downto 0);
signal keycode : std_logic_vector(3 downto 0);
signal deb_press : std_logic;
signal sh_reg : std_logic_vector(23 downto 0);
--*****************************************
component debounce
PORT(
        fs:IN STD_LOGIC;
        bounce_in:in std_logic;
        deb_out:out STD_LOGIC
      );
end component;
--*****************************
begin
key_code:--按鍵偵測及編碼電路
process(clk1)
begin
  if clk1='1' and clk1'event then
        if y="0000" then      --    鍵盤 com 共點接 0
              if y_code="11" then
                    y_code<="00";
              else y_code<=y_code+1;
              end if;
        elsif y="0001" then
```

防彈跳開關電路宣告

```
            keycode<=y_code&"00";
        elsif y="0010" then
            keycode<=y_code&"01";
        elsif y="0100" then
            keycode<=y_code&"10";
        elsif y="1000" then
            keycode<=y_code&"11";
        end if;
  end if;
end process key_code;
--*****************************
key_scan:     --列掃描信號輸出電路
process(y_code)
begin
        case y_code is
        when "00" => x<="1000";
        when "01" => x<="0100";
        when "10" => x<="0010";
        when "11" => x<="0001";
        when others=>x<="0000";
        end case;
end process key_scan;

--*****************************
load_4word:
block
signal press : std_logic;
begin
press<=(y(0) or y(1) or y(2) or y(3));     --按鍵訊號偵測，未按下為 0 按下為 1

debounce_mod:debounce
        port map(fs=>clk1,bounce_in=>press,deb_out=>deb_press);

end block load_4word;
--*********************************************************
shift:process(clr,deb_press)
begin
        if clr='1' then
            sh_reg(23 downto 0)<="000000000000000000000000";    --將輸出清除為"000000" 6 個 0，24bit
        elsif deb_press='1' and deb_press'event then      --按鍵防彈跳正緣信號觸發
          if ena='0' then
            sh_reg(23 downto 0)<= sh_reg(19 downto 0) & keycode; --輸入個位數字(4bit)其餘位元向左移位 4bit
          end if;
```

呼叫防彈跳開關電路

```
        end if;
end process shift;
--****************************

Scan_seg7:
process(Clk1)
  variable Scan1 : std_logic_vector(2 downto 0):="000";
   begin
        wait until clk1='1' ;
        if (Scan1="000") then
                Scan_seg <= "100000" ;
                bcd <= sh_reg(23 downto 20) ;
        elsif (Scan1="001") then
                Scan_seg <= "010000" ;
                bcd <= sh_reg(19 downto 16);
        elsif (Scan1="010") then
                Scan_seg <= "001000" ;
                bcd <= sh_reg(15 downto 12);
        elsif (Scan1="011") then
                Scan_seg <= "000100" ;
                bcd <= sh_reg(11 downto 8);
        elsif (Scan1="100") then
                Scan_seg <= "000010" ;
                bcd <= sh_reg(7 downto 4);
        else Scan_seg <= "000001" ;
                bcd <= sh_reg(3 downto 0);
        end if ;
        if Scan1 >= "101" then Scan1 := "000" ;
        else Scan1 := Scan1 + 1 ;
        end if ;
end process scan_seg7 ;
--*********************************************************
with bcd select
  seg <= "1111110"    when "0000" ,
         "0110000"    when "0001" ,
         "1101101"    when "0010" ,
         "1111001"    when "0011" ,
         "0110011"    when "0100" ,
         "1011011"    when "0101" ,
         "0011111"    when "0110" ,
         "1110000"    when "0111" ,
         "1111111"    when "1000" ,
         "1110011"    when "1001" ,
         "1000000"    when "1010" ,
         "0100000"    when "1011" ,
         "0010000"    when "1100" ,
         "0001000"    when "1101" ,
         "0000100"    when "1110" ,
         "0000010"    when "1111" ,
         "1001111"    when others ;
    end key    ;
```

7 節顯示器掃瞄輸出訊號

Chip/ POF	Device	Input Pins	Output Pins	Bidir Pins	Memory Bits	Memory % Utilized	LCs	LCs % Utilized
keyin_4word								
	EPF10K10TC144-4	7	17	0	0	0 %	101	17 %
User Pins:		7	17	0				

Project Information d:\ex\keyin_4word_epf10k\keyin_4word.rpt

** PIN/LOCATION/CHIP ASSIGNMENTS **

User Assignments	Actual Assignments (if different)	Node Name
keyin_4word@126		clk1
keyin_4word@48		clr
keyin_4word@47		ena
keyin_4word@17		scan_seg0
keyin_4word@14		scan_seg1
keyin_4word@60		scan_seg2
keyin_4word@62		scan_seg3
keyin_4word@63		scan_seg4
keyin_4word@64		scan_seg5
keyin_4word@18		seg0
keyin_4word@19		seg1
keyin_4word@20		seg2
keyin_4word@21		seg3
keyin_4word@22		seg4
keyin_4word@23		seg5
keyin_4word@26		seg6
keyin_4word@89		x0
keyin_4word@90		x1
keyin_4word@91		x2
keyin_4word@92		x3
keyin_4word@98		y0
keyin_4word@97		y1
keyin_4word@96		y2
keyin_4word@95		y3

4x4 鍵盤掃描輸入 6 位數 VHDL(2)- keyin_combi_com0_10k.vhd (鍵盤輸入信號 com 點接低電位)

利用三個副程式組合成一個電路(component… port map…)

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
entity keyin_combi_com0_10k is
port(
        clr : in std_logic;
        ena : in std_logic;
        clk1: in std_logic;--(40HZ~250HZ)
        clk2: in std_logic;
        y : in std_logic_vector(3 downto 0);--Y  列
        x : out std_logic_vector(3 downto 0);--X
        scan_seg : out std_logic_vector(5 downto 0);
        seg : out std_logic_vector(0 to 6)) ;
end keyin_combi_com0_10k;
--****************************************
architecture key of keyin_combi_com0_10k is
signal deb_press : std_logic;
signal mdu_clk : std_logic;
signal sh_reg : std_logic_vector(23 downto 0);
signal      bcdcode   : std_logic_vector(3 downto 0);
--*****************************************
component debounce
PORT(
        fs:IN STD_LOGIC;
        bounce_in:in std_logic;
        deb_out:out STD_LOGIC
      );
end component;
--*****************************
component key2_com0
port(
        clk1: in std_logic;
        x : out std_logic_vector(3 downto 0);
        y : in std_logic_vector(3 downto 0);
        bcd  :out std_logic_vector(3 downto 0));
end component;
--*********************************
component mdu_n
port(
        fin:in std_logic;
```

防彈跳電路

4x4 鍵盤輸入讀取判斷電路

除以 N 的除頻電路

此程式所呼叫之副程式共有三個

```
        fout:buffer std_logic );
end component;
begin
--*********************************
keyin_mod:key2_com0
      port map(clk1,x,y,bcdcode);
--*****************************
seg_mdu:mdu_n
      port map(clk2,mdu_clk);
load_6word:
block
signal press : std_logic;
begin
press<=(y(0) or y(1) or y(2) or y(3));    --按鍵訊號偵測，未按下為 0 按下為 1

debounce_mod:debounce
      port map(fs=>clk1,bounce_in=>press,deb_out=>deb_press);

end block load_6word;

--*********************************************************
shift:process(clr,deb_press)
begin
      if clr='1' then
            sh_reg(23 downto 0)<="000000000000000000000000";
      elsif deb_press='1' and deb_press'event then
        if ena='0' then
            sh_reg(23 downto 0)<= sh_reg(19 downto 0) & bcdcode;
        end if;
      end if;
end process shift;
--*****************************

seg7:block    --七節顯示器掃描及顯示電路，用 block 區塊方式將電路做區隔
signal bcd      :      std_logic_vector(3 downto 0);
begin
Scan_seg7:
process(mdu_clk)
  variable Scan1 : std_logic_vector(2 downto 0):="000";
    begin
      wait until mdu_clk='1' ;
      if (Scan1="000") then
              Scan_seg <= "100000" ;
```

呼叫 4x4 鍵盤輸入副程式，產生 bcdcode 鍵盤掃描資訊之編碼

--除 N 之除頻電路呼叫，產生 mdu_clk 除以 N 後之除頻訊號

呼叫防彈跳開關電路，產生 deb_press 之防彈跳訊號，給予移位暫存器當做觸發訊號

```
            bcd <= sh_reg(23 downto 20) ;
        elsif (Scan1="001") then
            Scan_seg <= "010000" ;
            bcd <= sh_reg(19 downto 16);
        elsif (Scan1="010") then
            Scan_seg <= "001000" ;
            bcd <= sh_reg(15 downto 12);
        elsif (Scan1="011") then
            Scan_seg <= "000100" ;
            bcd <= sh_reg(11 downto 8);
        elsif (Scan1="100") then
            Scan_seg <= "000010" ;
            bcd <= sh_reg(7 downto 4);
        else Scan_seg <= "000001" ;
            bcd <= sh_reg(3 downto 0);
        end if ;
        if Scan1 >= "101" then Scan1 := "000" ;
        else Scan1 := Scan1 + 1 ;
        end if ;
end process scan_seg7 ;
--*******************************************************
with bcd select
  seg <= "1111110"    when "0000" ,
         "0110000"    when "0001" ,
         "1101101"    when "0010" ,
         "1111001"    when "0011" ,
         "0110011"    when "0100" ,
         "1011011"    when "0101" ,
         "0011111"    when "0110" ,
         "1110000"    when "0111" ,
         "1111111"    when "1000" ,
         "1110011"    when "1001" ,
         "1000000"    when "1010" ,
         "0100000"    when "1011" ,
         "0010000"    when "1100" ,
         "0001000"    when "1101" ,
         "0000100"    when "1110" ,
         "0000010"    when "1111" ,
         "1001111"    when others ;
end block seg7;    --block 區塊結束
end key    ;
```

** DEVICE SUMMARY **

Chip/ POF	Device	Input Pins	Output Pins	Bidir Pins	Memory Bits	Memory % Utilized	LCs	LCs % Utilized
keyin_combi_com0_10k								
	EPF10K10TC144-4	8	17	0	0	0 %	121	21 %
User Pins:		8	17	0				

Project Information d:\ex\keyin_combi\keyin_combi_com0_10k.rpt

** PIN/LOCATION/CHIP ASSIGNMENTS **

User Assignments	Actual Assignments (if different)	Node Name
keyin_combi_com0_10k@126		clk1
keyin_combi_com0_10k@54		clk2
keyin_combi_com0_10k@48		clr
keyin_combi_com0_10k@47		ena
keyin_combi_com0_10k@17		scan_seg0
keyin_combi_com0_10k@14		scan_seg1
keyin_combi_com0_10k@60		scan_seg2
keyin_combi_com0_10k@62		scan_seg3
keyin_combi_com0_10k@63		scan_seg4
keyin_combi_com0_10k@64		scan_seg5
keyin_combi_com0_10k@18		seg0
keyin_combi_com0_10k@19		seg1
keyin_combi_com0_10k@20		seg2
keyin_combi_com0_10k@21		seg3
keyin_combi_com0_10k@22		seg4
keyin_combi_com0_10k@23		seg5
keyin_combi_com0_10k@26		seg6
keyin_combi_com0_10k@89		x0
keyin_combi_com0_10k@90		x1
keyin_combi_com0_10k@91		x2
keyin_combi_com0_10k@92		x3
keyin_combi_com0_10k@98		y0
keyin_combi_com0_10k@97		y1
keyin_combi_com0_10k@96		y2
keyin_combi_com0_10k@95		y3

4x4 鍵盤掃描輸入 6 位數 VHDL(2)- keyin_combi_com0_10k.vhd 所呼叫之副程式

(1)key2_com0.vhd 鍵盤掃描編碼電路程式

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
entity key2_com0 is
port(
	clk1: in std_logic;
	x : out std_logic_vector(3 downto 0);
	y : in std_logic_vector(3 downto 0);
	bcd  :buffer std_logic_vector(3 downto 0));
end key2_com0;
architecture key of key2_com0 is
signal scan	:	integer range 3 downto 0;
--*****************************************
component debounce
PORT(
	 fs:IN STD_LOGIC;
	 bounce_in:in std_logic;
	 deb_out:out STD_LOGIC
	);
end component;
--****************************
begin

key_scan:
process(clk1)
	begin
	 wait until clk1='1' ;
	if y="0000" then
		if scan=0 then
			scan<=3;
		else scan<=scan-1;
		end if;
	end if;
end process key_scan;
key_x_scan:
process(clk1)
begin
	case scan is
		when 3 => x<="1000";
		when 2 => x<="0100";
```

```
            when 1 => x<="0010";
            when 0 => x<="0001";
            when others=> null;
        end case;
end process key_x_scan;
--*************************
key_read:
process(y)
begin
if scan=3 then
        case y is
                when "1000" => bcd<="0011";
                when "0100" => bcd<="0010";
                when "0010" => bcd<="0001";
                when "0001" => bcd<="0000";
                when others => null;
            end case;
        elsif scan=2 then
          case y is
                when "1000" => bcd<="0111";
                when "0100" => bcd<="0110";
                when "0010" => bcd<="0101";
                when "0001" => bcd<="0100";
                when others => null;
            end case;
        elsif scan=1 then
          case y is
                when "1000" => bcd<="1011";
                when "0100" => bcd<="1010";
                when "0010" => bcd<="1001";
                when "0001" => bcd<="1000";
                when others => null;
            end case;
        elsif scan=0 then
          case y is
                when "1000" => bcd<="1111";
                when "0100" => bcd<="1110";
                when "0010" => bcd<="1101";
                when "0001" => bcd<="1100";
                when others => null;
            end case;
        end if ;
end process key_read;
```

```
--****************************
   end key    ;
```

4x4 鍵盤掃描輸入 6 位數 VHDL(2)- keyin_combi_com0_10k.vhd 所呼叫之副程式

(2)mdu_n.vhd----N 模之除頻電路

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity mdu_n is
generic ( N : integer := 9999 ) ;
port(
        fin:in std_logic;
        fout:buffer std_logic
       );
end mdu_n;
architecture beh of mdu_n is
begin
   process(fin)
     variable cnt:integer range N downto 0;
   begin
     if fin='1' and fin'event then
        if cnt=0 then
             fout<=not fout;
             cnt:=N;
          else
             cnt:=cnt-1;
          end if;
       end if;
   end process;
end beh;
```