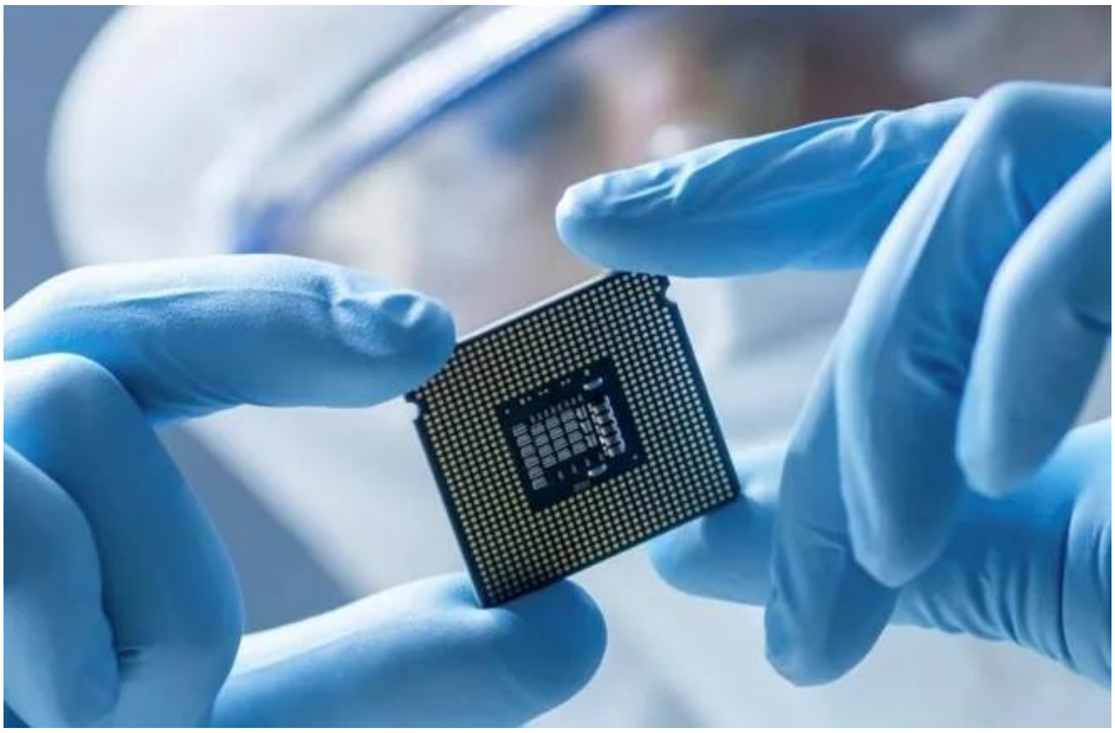


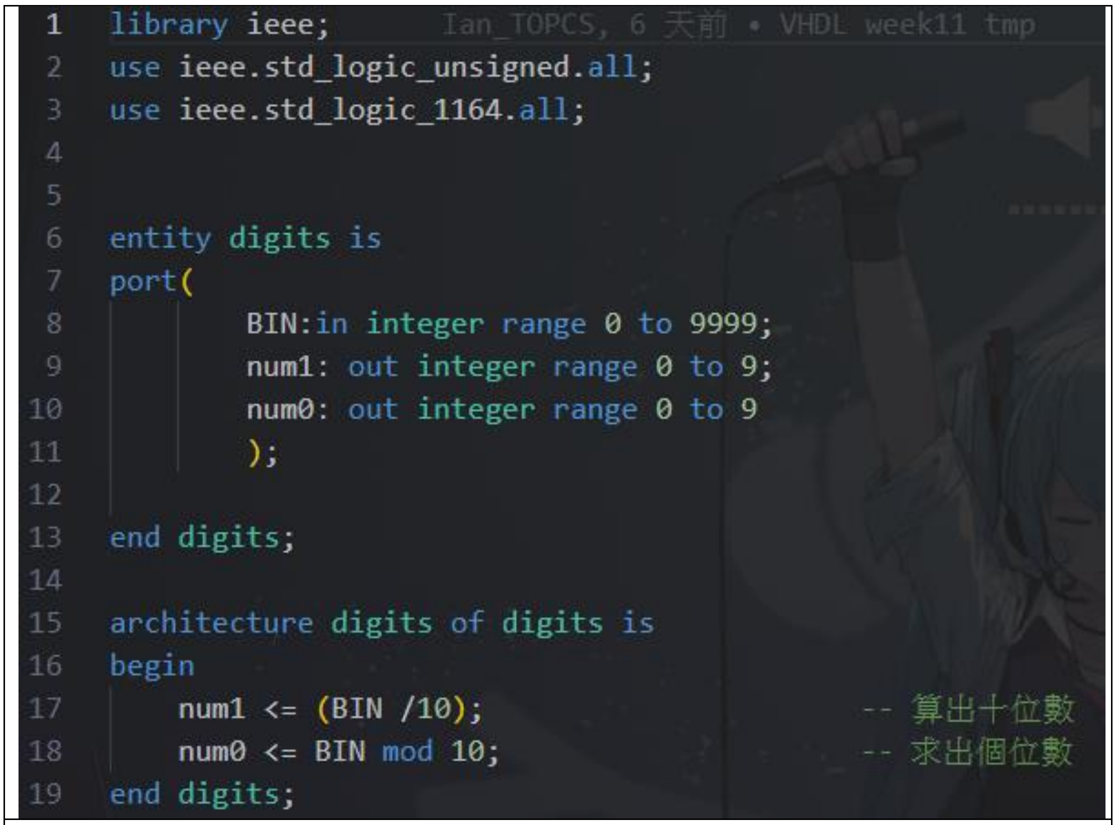
數位系統設計作業

HW9

學號 : 01257027 姓名 : 林承羿

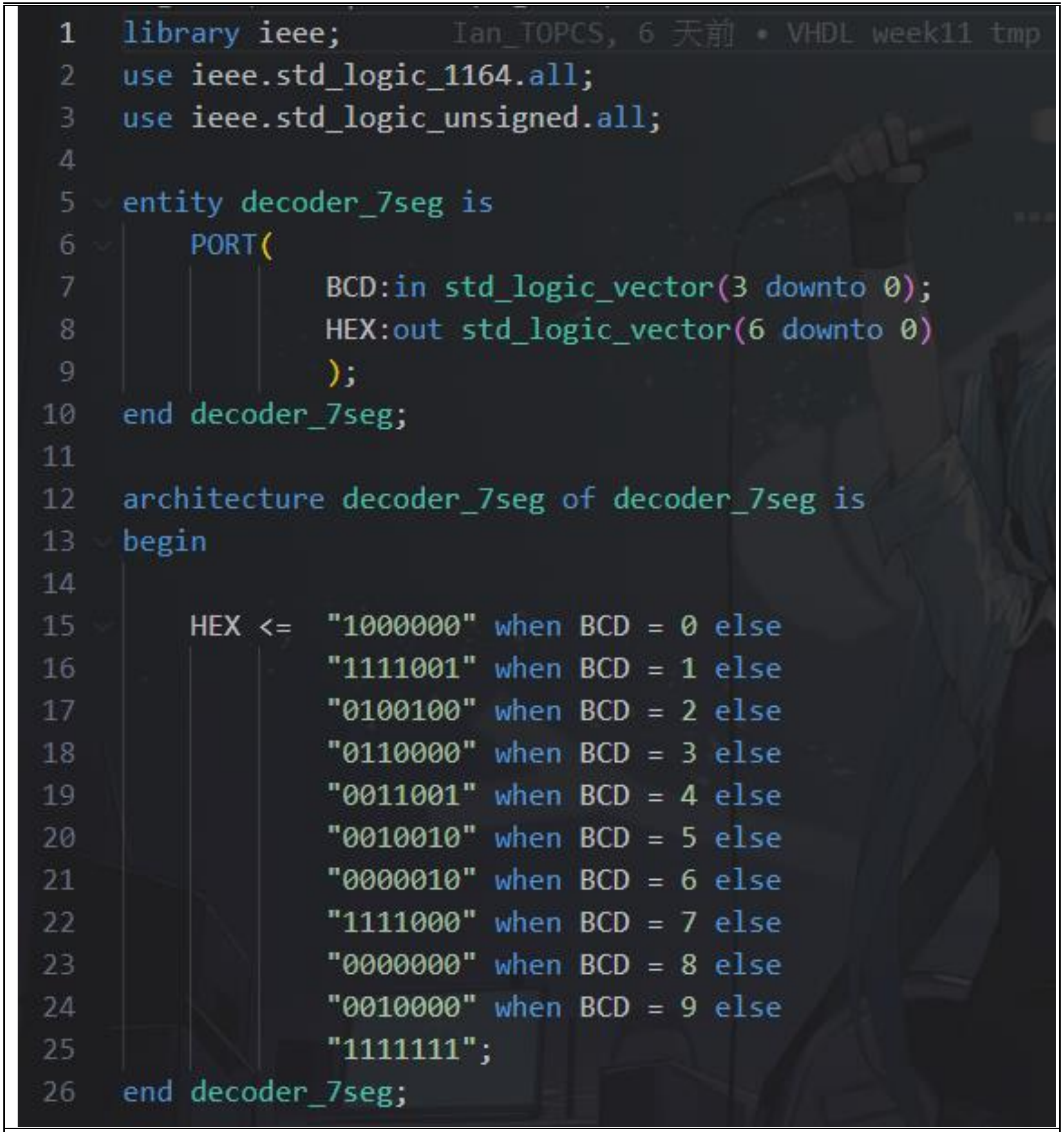
第一題

前置作業



```
library ieee;          Ian_TOPCS, 6 天前 • VHDL week11 tmp
use ieee.std_logic_unsigned.all;
use ieee.std_logic_1164.all;


entity digits is
port(
        BIN:in integer range 0 to 9999;
        num1: out integer range 0 to 9;
        num0: out integer range 0 to 9
        );

end digits;

architecture digits of digits is
begin
    num1 <= (BIN /10);                              -- 算出十位數
    num0 <= BIN mod 10;                             -- 求出個位數
end digits;
```

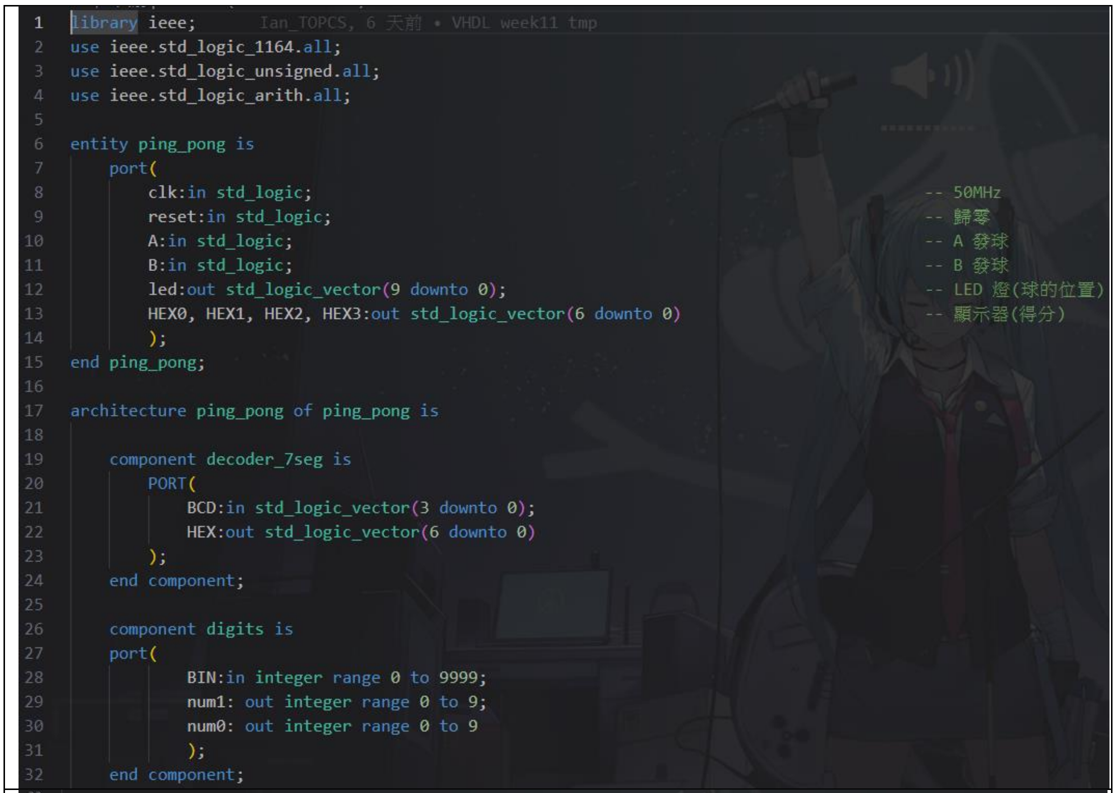
將分數求出 十位數 與 個位數



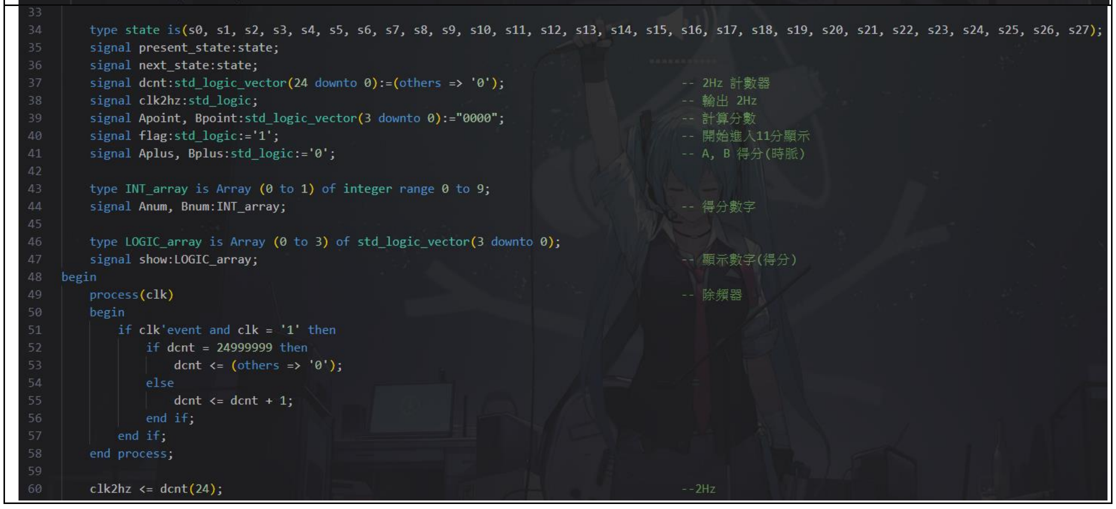
```vhdl
library ieee;        Ian_TOPCS, 6 天前 • VHDL week11 tmp
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity decoder_7seg is
    PORT(
            BCD:in std_logic_vector(3 downto 0);
            HEX:out std_logic_vector(6 downto 0)
            );
end decoder_7seg;

architecture decoder_7seg of decoder_7seg is
begin

    HEX <=  "1000000" when BCD = 0 else
            "1111001" when BCD = 1 else
            "0100100" when BCD = 2 else
            "0110000" when BCD = 3 else
            "0011001" when BCD = 4 else
            "0010010" when BCD = 5 else
            "0000010" when BCD = 6 else
            "1111000" when BCD = 7 else
            "0000000" when BCD = 8 else
            "0010000" when BCD = 9 else
            "1111111";
end decoder_7seg;
```

將十位數與各位數分別帶入七段顯示器

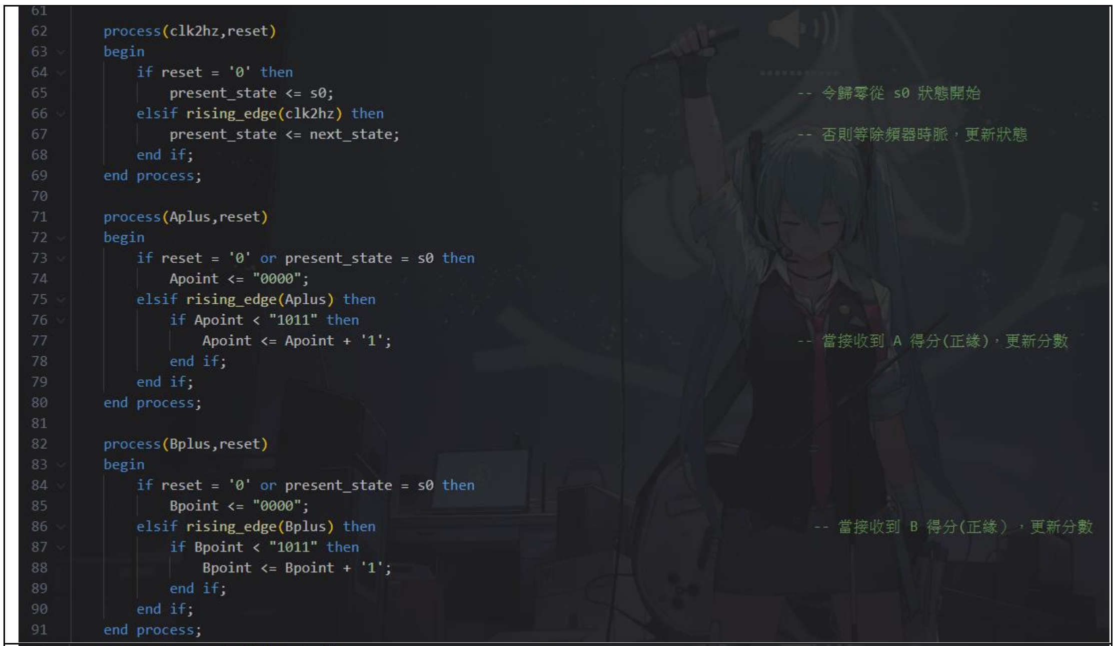
程式碼



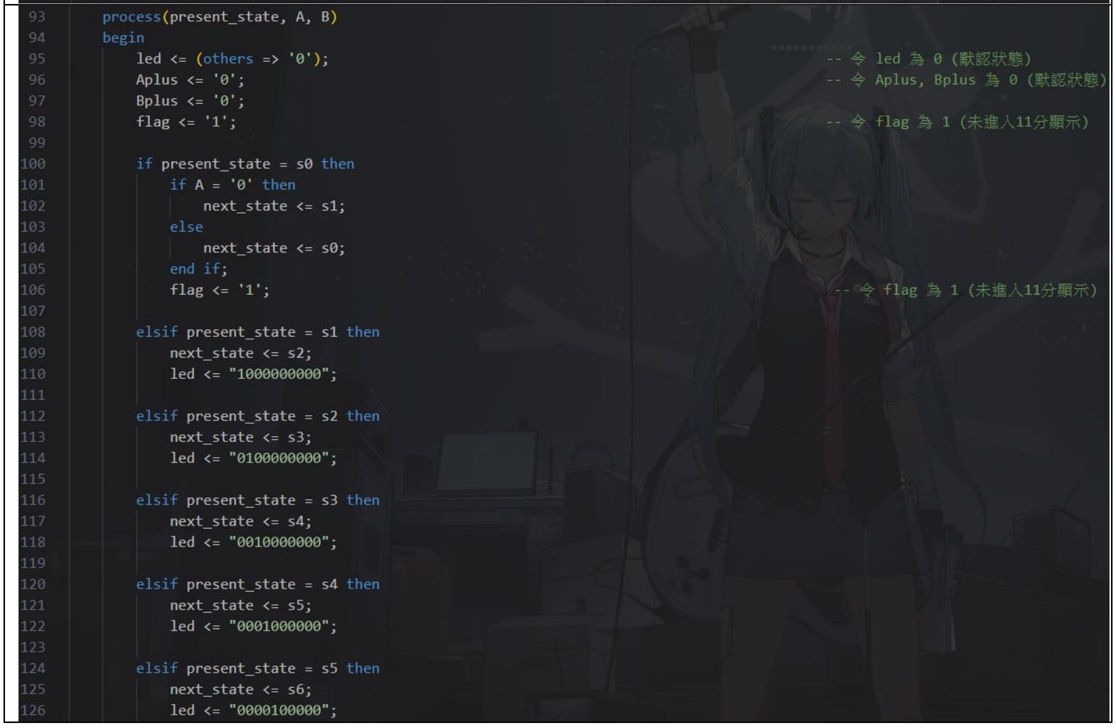
```vhdl
library ieee;      Ian_TOPCS, 6 天前 • VHDL week11 tmp
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity ping_pong is
    port(
        clk:in std_logic;                                                               -- 50MHz
        reset:in std_logic;                                                             -- 歸零
        A:in std_logic;                                                                 -- A 發球
        B:in std_logic;                                                                 -- B 發球
        led:out std_logic_vector(9 downto 0);                                           -- LED 燈(球的位置)
        HEX0, HEX1, HEX2, HEX3:out std_logic_vector(6 downto 0)                         -- 顯示器(得分)
        );
end ping_pong;

architecture ping_pong of ping_pong is

    component decoder_7seg is
        PORT(
            BCD:in std_logic_vector(3 downto 0);
            HEX:out std_logic_vector(6 downto 0)
        );
    end component;

    component digits is
    port(
            BIN:in integer range 0 to 9999;
            num1: out integer range 0 to 9;
            num0: out integer range 0 to 9
            );
    end component;
```



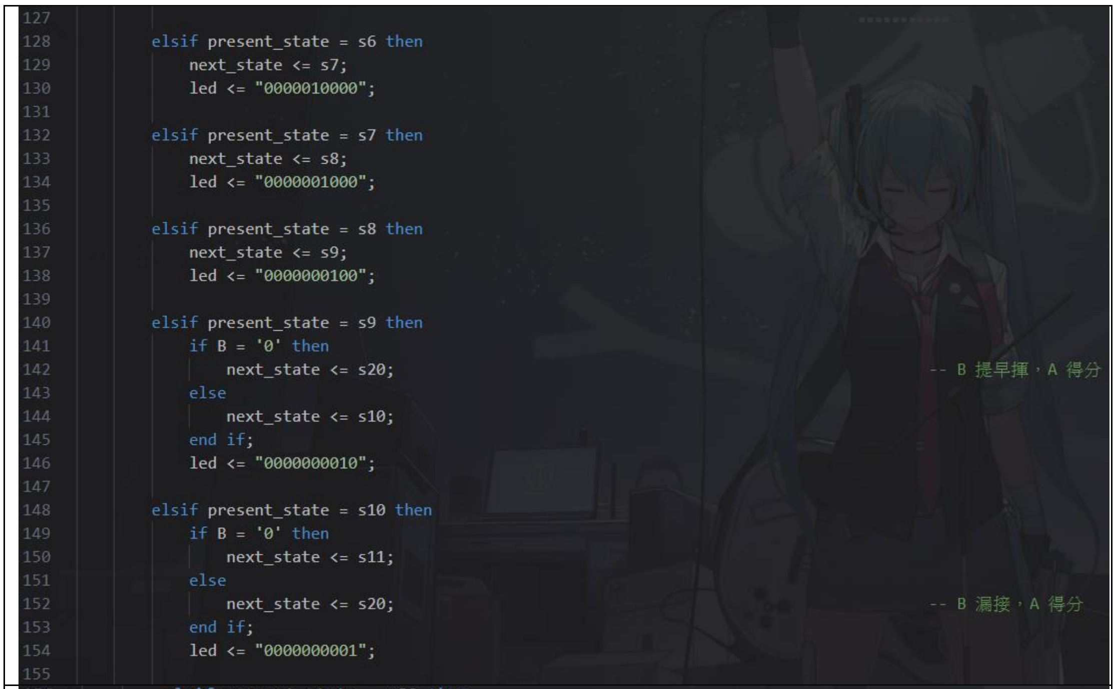
```vhdl

    type state is(s0, s1, s2, s3, s4, s5, s6, s7, s8, s9, s10, s11, s12, s13, s14, s15, s16, s17, s18, s19, s20, s21, s22, s23, s24, s25, s26, s27);
    signal present_state:state;
    signal next_state:state;
    signal dcnt:std_logic_vector(24 downto 0):=(others => '0');                 -- 2Hz 計數器
    signal clk2hz:std_logic;                                                    -- 輸出 2Hz
    signal Apoint, Bpoint:std_logic_vector(3 downto 0):="0000";                 -- 計算分數
    signal flag:std_logic:='1';                                                 -- 開始進入11分顯示
    signal Aplus, Bplus:std_logic:='0';                                         -- A, B 得分(時脈)

    type INT_array is Array (0 to 1) of integer range 0 to 9;
    signal Anum, Bnum:INT_array;                                                -- 得分數字

    type LOGIC_array is Array (0 to 3) of std_logic_vector(3 downto 0);
    signal show:LOGIC_array;                                                    -- 顯示數字(得分)
begin
    process(clk)                                                                -- 除頻器
    begin
        if clk'event and clk = '1' then
            if dcnt = 24999999 then
                dcnt <= (others => '0');
            else
                dcnt <= dcnt + 1;
            end if;
        end if;
    end process;

    clk2hz <= dcnt(24);                                                         --2Hz
```



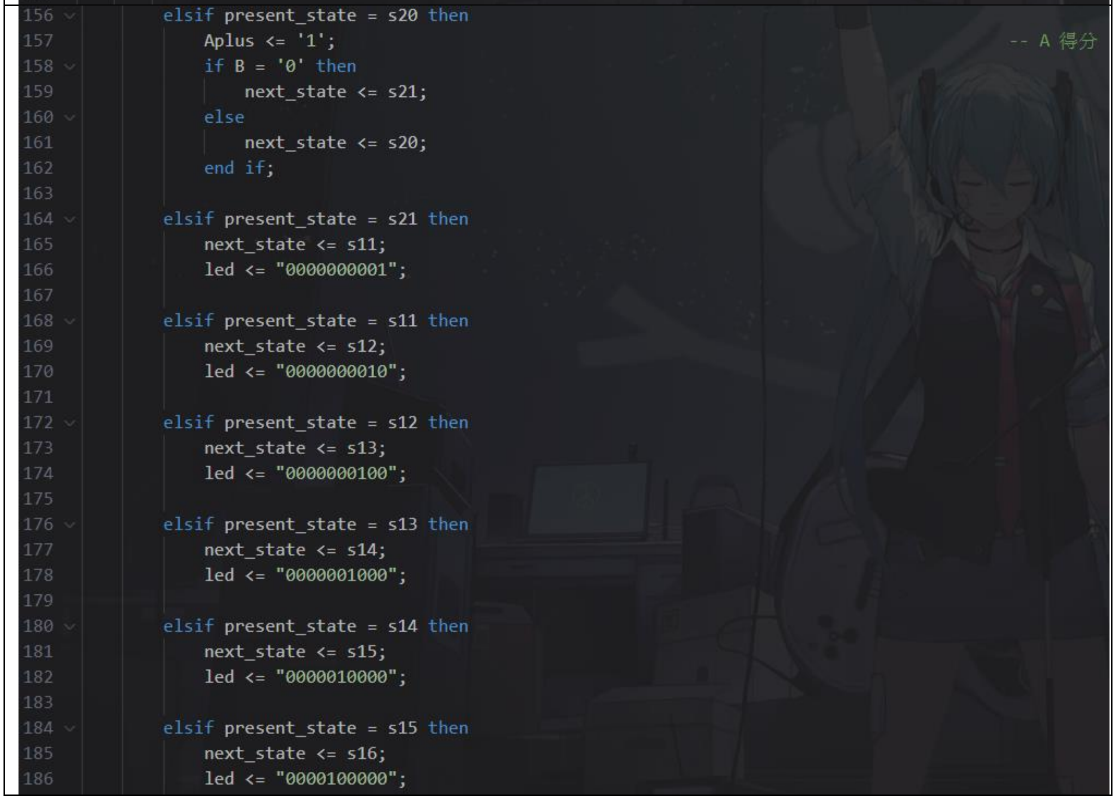
```vhdl

    process(clk2hz,reset)
    begin
        if reset = '0' then
            present_state <= s0;                                           -- 令歸零從 s0 狀態開始
        elsif rising_edge(clk2hz) then
            present_state <= next_state;                                   -- 否則等除頻器時脈，更新狀態
        end if;
    end process;

    process(Aplus,reset)
    begin
        if reset = '0' or present_state = s0 then
            Apoint <= "0000";
        elsif rising_edge(Aplus) then
            if Apoint < "1011" then
                Apoint <= Apoint + '1';                                    -- 當接收到 A 得分(正緣)，更新分數
            end if;
        end if;
    end process;

    process(Bplus,reset)
    begin
        if reset = '0' or present_state = s0 then
            Bpoint <= "0000";
        elsif rising_edge(Bplus) then                                       -- 當接收到 B 得分(正緣），更新分數
            if Bpoint < "1011" then
                Bpoint <= Bpoint + '1';
            end if;
        end if;
    end process;
```



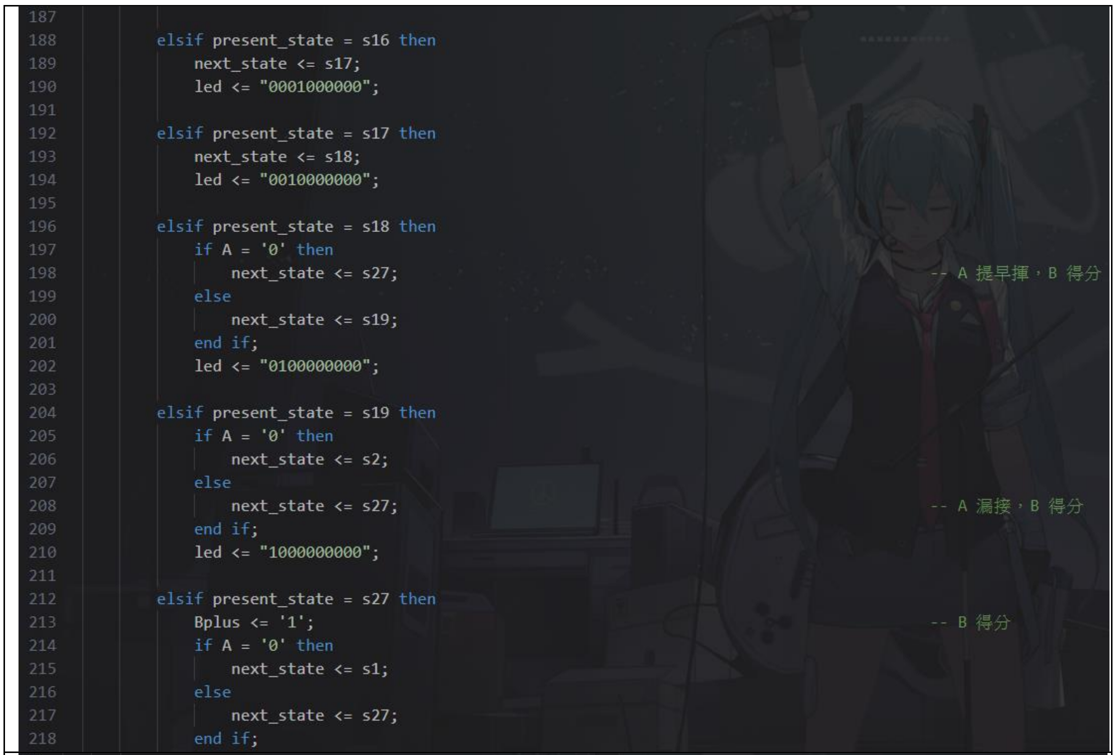
```vhdl
    process(present_state, A, B)
    begin
        led <= (others => '0');                                                -- 令 led 為 0 (默認狀態)
        Aplus <= '0';                                                          -- 令 Aplus, Bplus 為 0 (默認狀態)
        Bplus <= '0';
        flag <= '1';                                                           -- 令 flag 為 1 (未進入11分顯示)

        if present_state = s0 then
            if A = '0' then
                next_state <= s1;
            else
                next_state <= s0;
            end if;
            flag <= '1';                                                        -- 令 flag 為 1 (未進入11分顯示)

        elsif present_state = s1 then
            next_state <= s2;
            led <= "1000000000";

        elsif present_state = s2 then
            next_state <= s3;
            led <= "0100000000";

        elsif present_state = s3 then
            next_state <= s4;
            led <= "0010000000";

        elsif present_state = s4 then
            next_state <= s5;
            led <= "0001000000";

        elsif present_state = s5 then
            next_state <= s6;
            led <= "0000100000";
```



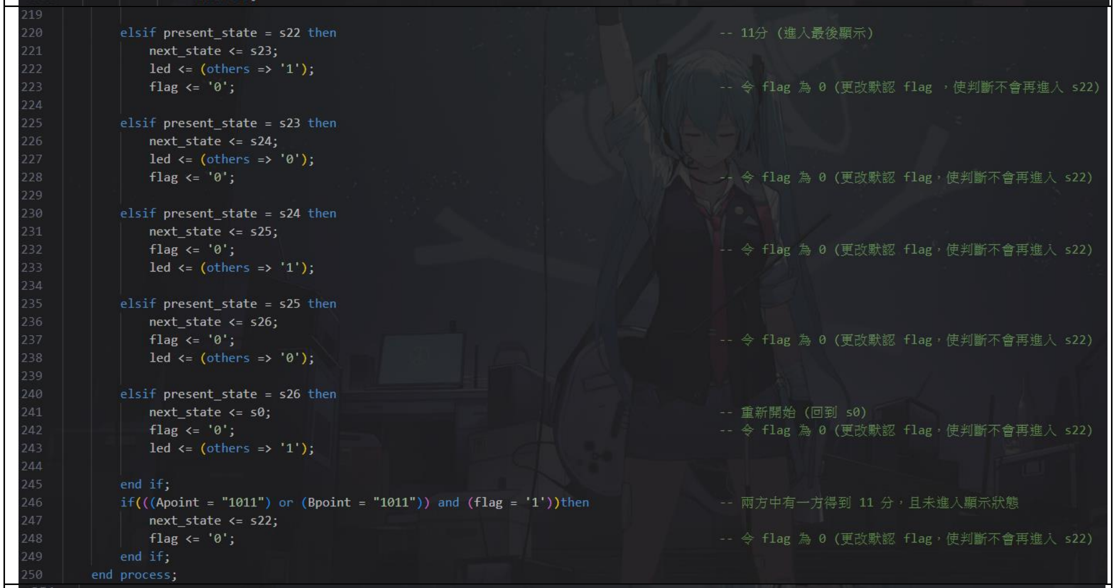
```vhdl

        elsif present_state = s6 then
            next_state <= s7;
            led <= "0000010000";

        elsif present_state = s7 then
            next_state <= s8;
            led <= "0000001000";

        elsif present_state = s8 then
            next_state <= s9;
            led <= "0000000100";

        elsif present_state = s9 then
            if B = '0' then
                next_state <= s20;                          -- B 提早揮，A 得分
            else
                next_state <= s10;
            end if;
            led <= "0000000010";

        elsif present_state = s10 then
            if B = '0' then
                next_state <= s11;
            else
                next_state <= s20;                          -- B 漏接，A 得分
            end if;
            led <= "0000000001";

```



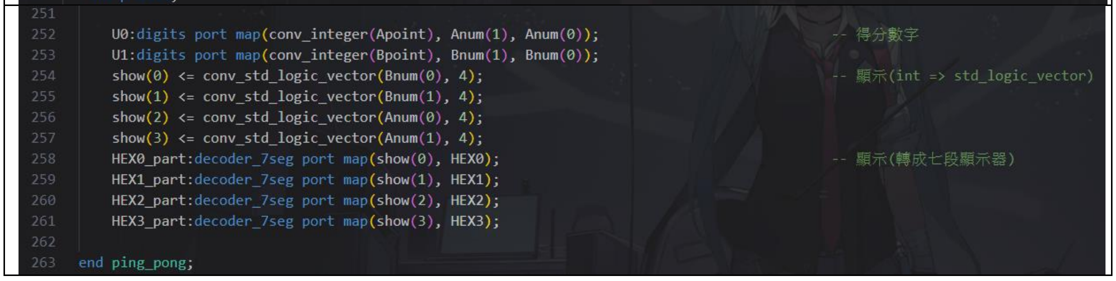
```vhdl
        elsif present_state = s20 then
            Aplus <= '1';                                   -- A 得分
            if B = '0' then
                next_state <= s21;
            else
                next_state <= s20;
            end if;

        elsif present_state = s21 then
            next_state <= s11;
            led <= "0000000001";

        elsif present_state = s11 then
            next_state <= s12;
            led <= "0000000010";

        elsif present_state = s12 then
            next_state <= s13;
            led <= "0000000100";

        elsif present_state = s13 then
            next_state <= s14;
            led <= "0000001000";

        elsif present_state = s14 then
            next_state <= s15;
            led <= "0000010000";

        elsif present_state = s15 then
            next_state <= s16;
            led <= "0000100000";
```



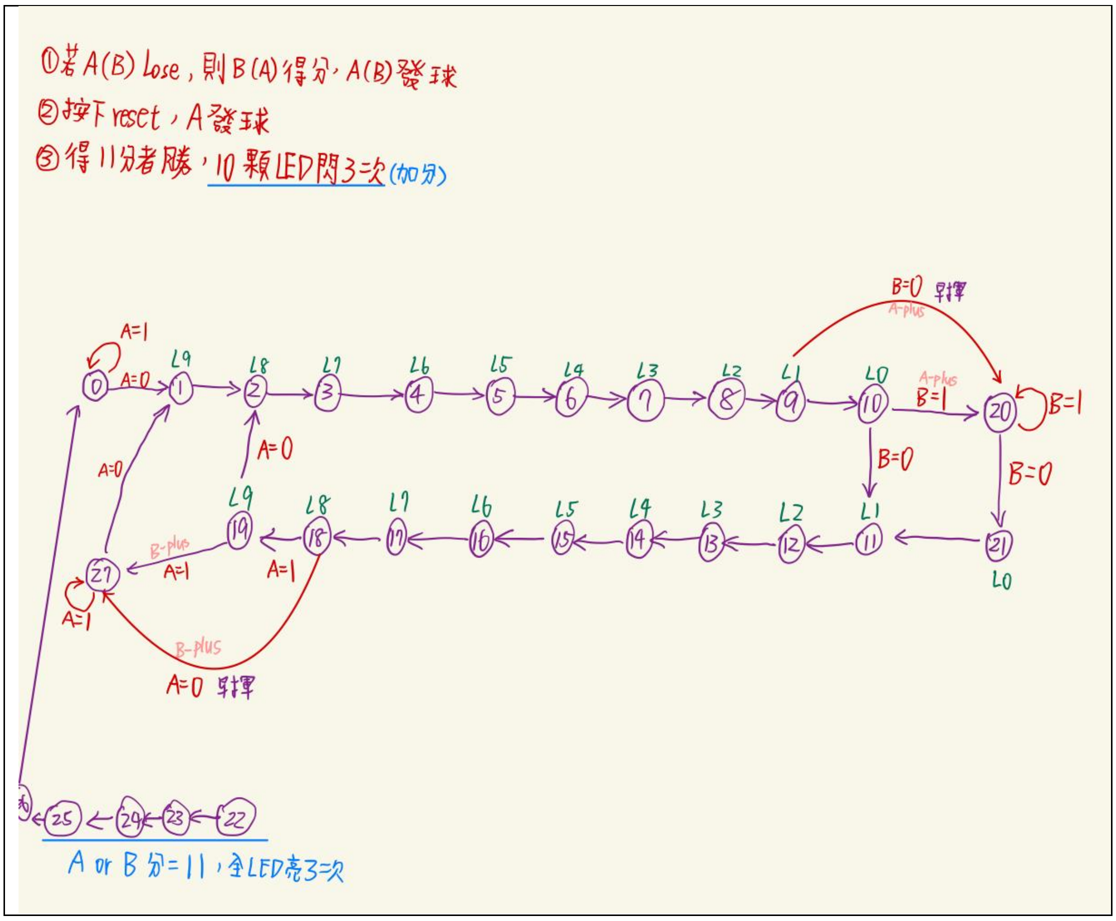
```vhdl

        elsif present_state = s16 then
            next_state <= s17;
            led <= "0001000000";

        elsif present_state = s17 then
            next_state <= s18;
            led <= "0010000000";

        elsif present_state = s18 then
            if A = '0' then
                next_state <= s27;                          -- A 提早揮，B 得分
            else
                next_state <= s19;
            end if;
            led <= "0100000000";

        elsif present_state = s19 then
            if A = '0' then
                next_state <= s2;
            else
                next_state <= s27;                          -- A 漏接，B 得分
            end if;
            led <= "1000000000";

        elsif present_state = s27 then
            Bplus <= '1';                                   -- B 得分
            if A = '0' then
                next_state <= s1;
            else
                next_state <= s27;
            end if;
```



```vhdl

        elsif present_state = s22 then                      -- 11分 (進入最後顯示)
            next_state <= s23;
            led <= (others => '1');
            flag <= '0';                                    -- 令 flag 為 0 (更改默認 flag ，使判斷不會再進入 s22)

        elsif present_state = s23 then
            next_state <= s24;
            led <= (others => '0');
            flag <= '0';                                    -- 令 flag 為 0 (更改默認 flag，使判斷不會再進入 s22)

        elsif present_state = s24 then
            next_state <= s25;
            flag <= '0';                                    -- 令 flag 為 0 (更改默認 flag，使判斷不會再進入 s22)
            led <= (others => '1');

        elsif present_state = s25 then
            next_state <= s26;
            flag <= '0';                                    -- 令 flag 為 0 (更改默認 flag，使判斷不會再進入 s22)
            led <= (others => '0');

        elsif present_state = s26 then
            next_state <= s0;                               -- 重新開始 (回到 s0)
            flag <= '0';                                    -- 令 flag 為 0 (更改默認 flag，使判斷不會再進入 s22)
            led <= (others => '1');

        end if;
        if(((Apoint = "1011") or (Bpoint = "1011")) and (flag = '1'))then   -- 兩方中有一方得到 11 分，且未進入顯示狀態
            next_state <= s22;
            flag <= '0';                                    -- 令 flag 為 0 (更改默認 flag，使判斷不會再進入 s22)
        end if;
    end process;
```



```vhdl

    U0:digits port map(conv_integer(Apoint), Anum(1), Anum(0));              -- 得分數字
    U1:digits port map(conv_integer(Bpoint), Bnum(1), Bnum(0));
    show(0) <= conv_std_logic_vector(Bnum(0), 4);                            -- 顯示(int => std_logic_vector)
    show(1) <= conv_std_logic_vector(Bnum(1), 4);
    show(2) <= conv_std_logic_vector(Anum(0), 4);
    show(3) <= conv_std_logic_vector(Anum(1), 4);
    HEX0_part:decoder_7seg port map(show(0), HEX0);                          -- 顯示(轉成七段顯示器)
    HEX1_part:decoder_7seg port map(show(1), HEX1);
    HEX2_part:decoder_7seg port map(show(2), HEX2);
    HEX3_part:decoder_7seg port map(show(3), HEX3);

end ping_pong;
```



心得

經過此次的實驗，我更了解何謂狀態圖的規劃，在每一次的狀態切換，都依然在自己預期的範圍內。在這一次作業中，我體悟最深的是何時要做分數的更改，如果在狀態 9、10 都做加分，可能因為偵測時根本還沒按，分數也會有所錯誤，故應該是在確定到某狀態後才加，除了保證結果的正確外，也讓自己觀念更加清楚。