

CPLD-Based 溫度控制 IC 晶片之設計與實現

摘 要

本論文主要以 VHDL(Very High Speed Integrated Circuit Hardware Description Language)硬體描述語言來設計，並以硬體 CPLD(Complex Programmable Logic Device)作整合之架構，實現溫度控制之設計與製作，並藉由 ALTERA 公司開發的 MAX+PLUS II 應用軟體，來模擬並驗證每個設計環節的正常性，並由 Compiler，再燒載在 FLEX 10K，作為溫度控制的功能；最後我們也可以(在功能上)作進一步的擴充，達到溫室(溫度、照度、濕度)自動化控制的功能。

目 次

- | |
|------------------|
| 一. 前言 |
| 二. 溫度控制系統架構及電路分析 |
| 三. 模糊控制 |
| 四. VHDL 溫度控制程式 |
| 五. 軟體模擬、驗證 |
| 六. 結論及未來展望 |
| 七. 參考文獻 |

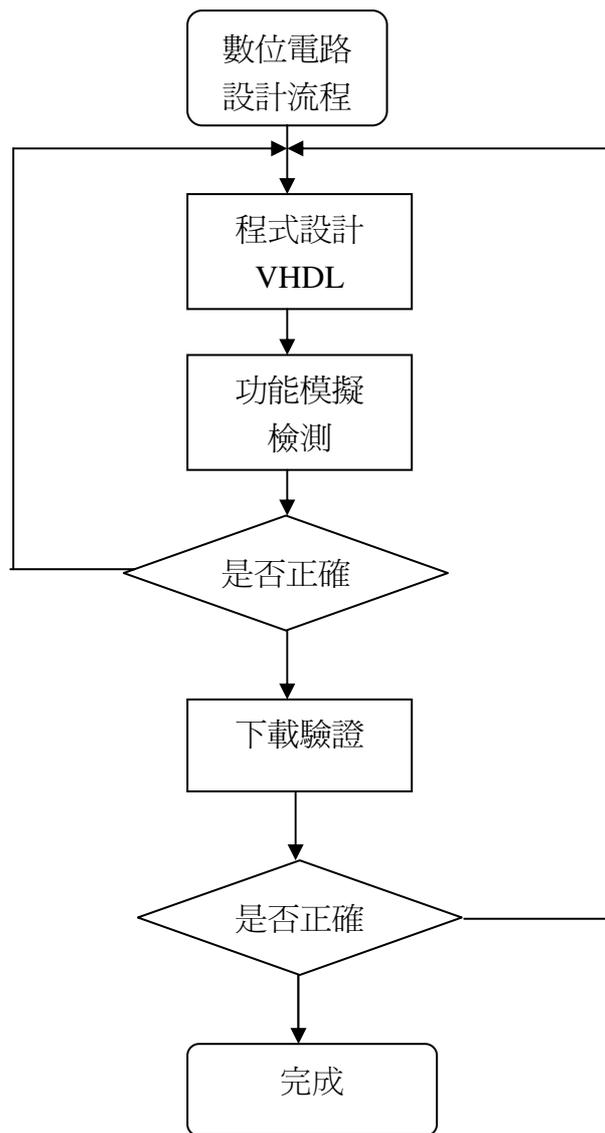
關鍵字：VHDL，CPLD

(一).前言：

e 世代數位電路的產品：

近年來資訊工業的快速發展，從日常用品中就能隨手可得；如通訊用的手機、電子錢幣的刷卡、數位相機、數位電視、電腦本身或其週邊，都是數位電路之產物。所以，我們看到半導體工業迅速的增長，而它賴以為生的大型數位積體電路(VLSI)就更顯示出它的重要性，因此，要融入這個潮流脈動中，數位電路的基本知識與經驗是不可或缺的，甚至還要具備了解設計架構稍大且完整電路的能力，才可跟得上時代的腳步。

以往電路設計，非要把全部的電子元件串接連線在一起不可，才算大功告成；若電路是小巧型的，還可以將就逐一完成，一旦電路非常龐大，就如同蜘蛛網，千頭萬緒理也理不清，其盤根錯節，真叫人徬徨失措，而布林代數化簡，縱然能化繁為簡，但一碰觸到超大型數位序向邏輯電路，那可就依然沒輒，真讓人難以施展；為了能改進以往數位電路設計繁複及佈線，並且縮短開發大型數位電路的時間，所以有了整合式數位電路設計環境的發展，此種環境是以個人電腦(PC)為平台，配合電子設計自動化(Electronic Design Automatic，EDA)軟體的執行，達到後電路設計輸入、模擬、下載驗證、修改、燒錄一氣呵成(圖一)，不僅讓寫程式作設計變得有效率，而且也讓自行設計開發邏輯晶片的夢想，得以實現。



圖一：數位電路設計流程

(二)、溫度控制系統架構及電路分析

$VR_1 + R_1 = 1.8K + 8.2K = 10K\Omega$ 時，則圖跨距調整電路的輸出電壓測試點 T_1 為 $10mV/K$ （即 $1\mu A/K \times 10k\Omega = 10mV/K$ ），當 AD590 感測到室溫

$25^\circ C (273.2 + 25 = 298.2^\circ K)$ 時，則 I_s 為 $298.2 \mu A$ ，則 T_1 的輸出電壓為 $298.2\mu A \cdot 10K = 2982mV = 2.982V$ 。

由於有些場合使用攝氏 ($^\circ C$) 單位較適合、方便,就如同一般我們常說的幾度 C ，因此圖下半部電路主要是用來將圖上半部電路的絕對溫度 ($^\circ K$) 為單位的輸出電壓值再加以轉換為以 $^\circ C$ 單位的電路，其輸出電壓變化為 $100mV/^\circ C$ 。

AD590 的輸出端接 ADC0809 的 INO 腳，將溫度類比的感測值轉為 8 位元數位電壓信號 ($2^{-1} \sim 2^{-8}$)，此信號接到 CPLD FLEX10K 的 DATA0~DATA7 之輸入腳，經模糊控制寫成之 VHDL 程式，在 temp 端輸出一信號控制 PLANT 受控體，形成溫度控制 IC 之功能。

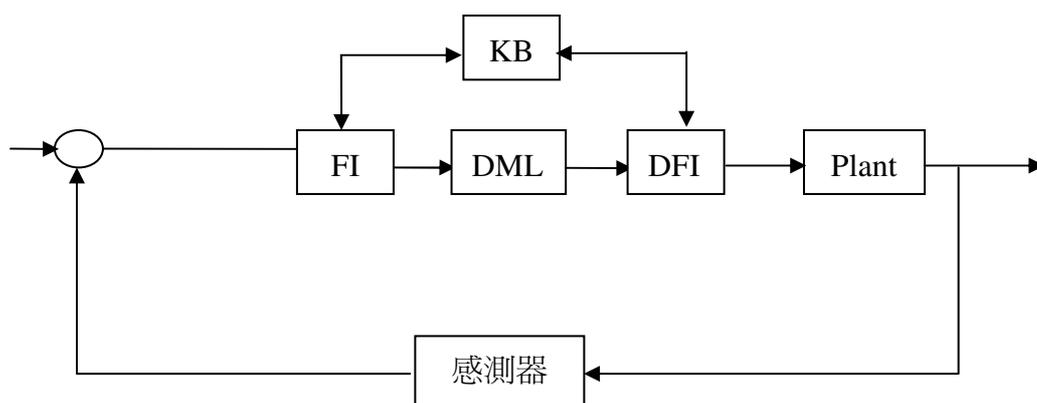
(三)、模糊控制

模糊控制之特色係根據不清晰的訊息，透過近似推理的過程而得到正確的結果，與人腦中「過程模糊，結論明確」的思維特徵類似，故利用模糊理論，將人類的推理、決策、判斷、控制等知識思考，轉化成知識庫及規則庫儲存於電腦中，再經由模糊推論法 (fuzzy inference) 以數值計算法完成推論的過程。本篇論文將以單輸入單輸出模糊器微例說明其原理及如何以 CPLD 設計與實現。

模糊控制器

傳統控制器的設計，通常須要數學模型來描述受控系統；但對於龐大、非線性或難以量測等情況的受控系統，考慮的變數既多且處理的問題複雜，所以要建立受控體 (plant) 的數學模式也就更困難，造成了設計及使用上的限制。

模糊控制係以系統的操作法則代替系統作精確的數學模型描述；模糊控制經由系統輸出回授，反覆對系統作誤差修正，以得到合乎要求的控制效果。事實上，模糊控制係建立在人類經驗的基礎上，換言之，人類並不是經由精確的數學表示式去瞭解受控系統，而是靠經驗及直覺做適當的控制；此控制法可視為一種探索式的決策規則，若將人類的經驗加以總結和描述，應用語言表達成一定性的條件語句及所需的決策規則，然後利用模糊數學為工具使其定量化，進而設計一控制器來實踐人類經驗，即可模仿人類的思維方式及操作策略，使模糊控制器能代替人類對複雜的控制系統進行控制。



圖三:為模糊控制圖。

本專題所設計的是一個單輸入單輸出之模糊控制器，此模糊控制器主要可分成：模糊化單元、決策邏輯單元、知識規則庫及解模擬化單元四部分。

模糊化介面 (FI)

模糊化過程是模糊控制器設計的第一步驟，其功能類似數位控制系統中的 A/D 轉換器；模糊化 (fuzzification) 乃是一種將外在輸入化為主觀意識的表達型態，其推衍的結果不在是單一結論值，而是依模糊概念衍生之多重推論值，此主觀意識以模糊歸屬函數實現各模糊集合，由此可瞭解到模糊化會影響輸入對控制器所代表的意義，也會對控制器產生的輸出結果造成影響，故歸屬函數的定義往往會造成推論控制力是否能完善控制系統之重要指標。對不同系統而言，由於輸入輸出特性不同，因此歸屬函數的定義有所不同；若是同一系統，定義的歸屬函數不同，所得到的控制結果也不同。因此為求得最佳控制效果，近年來不少學者專家試著以各種不同學習機制去訓練出最理想的歸屬函數之形狀，如基因遺傳演算法、類神經網路與模擬退火演算法等。

模糊化方式一般可分為模糊單值法 (fuzzy singleton) 及模糊數法 (fuzzy number)，其推論技巧各有不同，效果亦互異。

本論文之模糊化係以單值法作為設計考量

模糊單值 (fuzzy singleton)：

對大多數的模糊控制器而言，由量測儀器所得的輸入資料是明確值，所以必須把輸入資料轉化為明確的模糊單值 (fuzzy singleton)，由式 3.1 可知，當輸入 $x = x_0$ 時的輸出為單值 1，在其餘的點則為零。

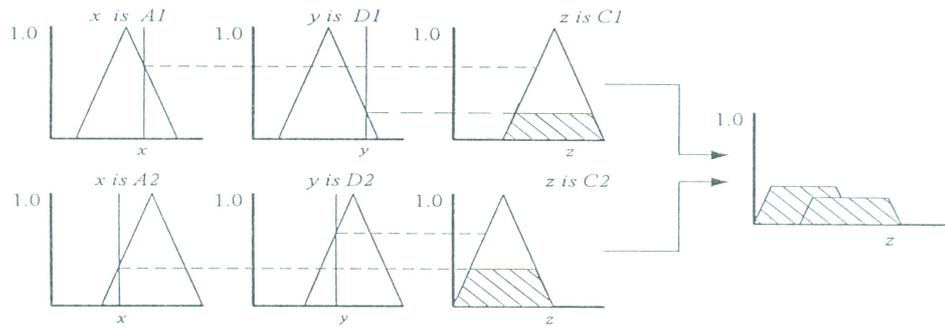
$$\mu_A(x) = \begin{cases} 0; & x \neq x_0 \\ 1; & x = x_0 \end{cases} \quad (3.1)$$

決策邏輯單元 (DML)

決策邏輯是模糊邏輯控制器的主要處理單元與核心，其具有模擬人類制定決策的能力。決策邏輯可由輸入資料經知識庫判斷決定輸出的對應值，它主要利用模糊蘊含 (fuzzy implication) 和模糊推論的合成運算 (compositional operators) 推導出模糊控制律。

模糊推論方法是建立在由 Zadeh 在 1973 所提出的近似推理方法上，而模糊推論方法有很多種，例如 Mamadani 的 MAX-MIN 模糊推論法，Lasern 的 MAX-Product 的模糊推論方法，Tsukamoto 的模糊推論方法，Takagi 和 Sugeno 的推論方法等等。而本論文，將介紹本論文所應用的 Mamadani 的 MAX-MIN 模糊推論方法。

(1) Mamadani 的 MAX-MIN 模糊推論法



圖四 MAX-MIN 模糊推論法

MAX-MIN 法的運算過程可以由圖四觀察出，而其他整個運算過程的步驟列述如下：

(a) 計算每一條規則的前提適合度：

$$w_i = \min\{\min(A_i, x), \min(D_i, y)\} \quad (3.2)$$

i : 每條規則的編號 x, y : 輸入值

兩個輸入變數分別與對應的前提項歸屬函數找出交集的點，再取兩點最小值。

(b) 計算每一條規則的結論項適合度：

$$B_i = \min(w_i, \mu_{B_i}(C)) \quad (3.3)$$

μ_{B_i} : 結論項的歸屬函數

(c) 結合所有規則的動作量：

$$B^* = \max_{i=1}^r B_i \quad (3.4)$$

r : 所觸發規則的數目

知識庫 (KB)

知識庫包含應用領域的知識及控制目標，可分為資料庫 (data base) 和語意控制規則庫 (rules base)。

資料庫的建立一般說來，可分為兩種方式建立，一為由設計者本身的經驗與專家知識所建造，使系統能貼切地使用於被控環境中；另一則是當系統的模型建立後，利用學習架構自我學習出規則庫，學習的方法可以利用類神經網路、模擬退火演算法或遺傳基因演算法去實現。

語意規則一般都以 IF 及 THEN 關係來實現，其所對應的歸屬函數設計有不同的形式，可依設計者的經驗或系統的性能要求加以選擇。

解模糊化 (DFI)

解模糊主要目的是要決定一個明確的輸出，將一個模糊控制動作對映到一個明確非模糊的控制動作，也就是將模糊推論的輸出，轉換成明確的數值，以作為受控系統之控制律。

目前解模糊化最常見的方法有：最大歸屬度法 (maximum criterion method, MC)、最大歸屬度法平均法 (mean of maximum method, MOM)、重心法 (center of area method, COA)、加權平均法 (center average weighting method)。

在本論文中採用此加權平均法來實現模糊晶片設計。式 (3.5) 為其數學表示式。

$$U^* = \frac{\sum y_i \cdot w_i}{\sum y_i} \quad (3.5)$$

w_i : 各推論得到之歸屬度

y_i : 其所觸發歸屬函數中心值

另外還有其他各種解模糊化方式，目的都在於將推論結果的平均值換算成實際的操作量。

(四)、溫度控制之 VHDL 程式：

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
entity temc is
port(clk:in std_logic;
      data:in std_logic_vector(7 downto 0);
      start,hz:out std_logic;
      ale,oen:out std_logic;
      add:out std_logic_vector(2 downto 0);
      temp:out std_logic;
      sg:out std_logic_vector(7 downto 0);
      com:out std_logic_vector(5 downto 0) );
end temc;
architecture a of temc is
signal osc:std_logic;
signal f:std_logic_vector(5 downto 0);
begin
  com(5 downto 0)<=f(5 downto 0);
  add<="000";
  oen<='1';
  process
  variable d_ff:std_logic_vector(20 downto 0);
  begin
    wait until clk='1';
    if d_ff(20 downto 0)>=2e6 then
      d_ff(20 downto 0):="00000000000000000000";
      else d_ff(20 downto 0):=d_ff+1;
    end if;
    hz<=not d_ff(6);
    osc<=not d_ff(8);
    start<=not d_ff(18);
  
```

```

        ale<=not d_ff(19);
    end process;
process(osc,clk)
variable d:std_logic_vector(3 downto 0);
variable sw:std_logic_vector(2 downto 0);
variable count:std_logic_vector(23 downto 0);
variable up:std_logic_vector(7 downto 0);
variable idp:std_logic_vector(8 downto 0);
begin
if (osc='1' and osc'event) then
    if (sw(2 downto 0)="110") then
        sw(2 downto 0):="000";
    else sw(2 downto 0):=sw(2 downto 0)+1;
    end if;
end if;
if (clk='1' and clk'event) then
    if (idp=data+"0101") then
        count(15 downto 8):=up;
        idp:="00000000";
        up:="00000000";
    else idp:=idp+1;
        count(15 downto 8):=count(15 downto 8);
    end if;
if (idp/=data+"0101") then
    if (up(3 downto 0)>="1010") then
        up(3 downto 0):="0000";
    else up(3 downto 0):=up(3 downto 0)+1;
    end if;
if (up(7 downto 4)="1010") then
    up(7 downto 4):="0000";
    elsif (up(3 downto 0)="1010") then
        up(7 downto 4):=up(7 downto 4)+1;
    end if;
else count(23 downto 16):="00000000";
end if;
    if (count(15 downto 8)>"00101000") then
        temp<='1';
    else temp<='0';
end if;
end if;
end if;

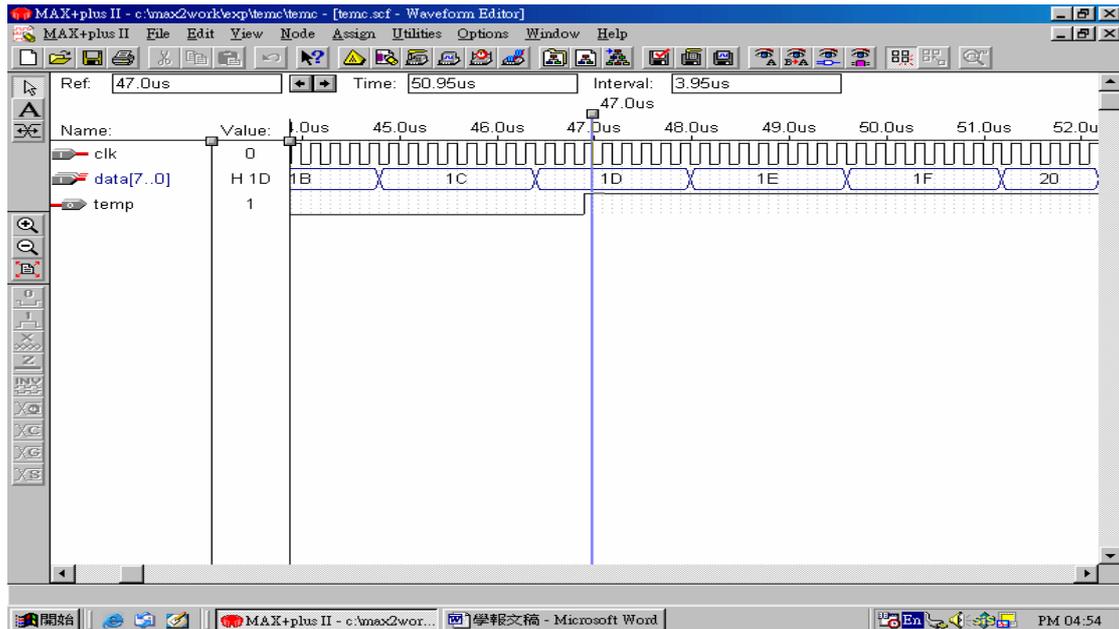
```

```

count(7 downto 0):="11111100";
case d is
  when"0000"=>sg<="00111111";
  when"0001"=>sg<="00000110";
  when"0010"=>sg<="01011011";
  when"0011"=>sg<="01001111";
  when"0100"=>sg<="01100110";
  when"0101"=>sg<="01101101";
  when"0110"=>sg<="01111101";
  when"0111"=>sg<="00000111";
  when"1000"=>sg<="01111111";
  when"1001"=>sg<="01100111";
  when others=>sg<="01110001";
end case;
case sw is
  when"000"=>d(3 downto 0):=count(3 downto 0);
    f(5 downto 0)<="111110";
  when"001"=>d(3 downto 0):=count(7 downto 4);
    f(5 downto 0)<="111101";
  when"010"=>d(3 downto 0):=count(11 downto 8);
    f(5 downto 0)<="111011";
  when"011"=>d(3 downto 0):=count(15 downto 12);
    f(5 downto 0)<="110111";
  when"100"=>d(3 downto 0):=count(19 downto 16);
    f(5 downto 0)<="101111";
  when"101"=>d(3 downto 0):=count(23 downto 20);
    f(5 downto 0)<="011111";
    when others=>d(3 downto 0):="1100";
      f(5 downto 0)<="111111";
end case;
end process;
end a;

```

(五) 模擬驗證：



(六) 結論與未來展望：

結論：

本篇論文是以一顆可程式邏輯元件 CPLD 設計完成之溫度控制器，它適用於溫室中之室溫控制，讓溫室裡植物能得到氣候的調節，而能茁壯長大。

此溫控設計完全由 VHDL 硬體描述語言來編寫，而由 CPLD 晶片來實現，如此一來，可簡化整體電路的結構，降低電路佈線的複雜性，對於整個電路的成本也可大幅的降低。

在 CPLD 內部所處理的均屬數位信號，而數位信號與類比信號相比，數位電路的輸出穩定、抗雜訊能力強，由模擬信號的驗證或觀察輸出信號的情形就可以馬上知道電路的設計是否正確，電路實現的過程中雖然經歷多次修改，但是 CPLD 猶如同同一張空白書紙，可由設計者任意自由發揮，不滿意可隨時擦掉重來，再加上 Altera 公司之開發軟體 MAX+plus II 10.0 功能之強大，讓使用者在整個設過程中並不會遭遇到很大的困難。最後完成整個數位式控制器的電路系統。在 A/D 的電路我們使用一個 8 位元的 IC(ADC0809)來實現，若能使用更多位元數的 IC 來實現，相信在溫度的靈敏度表現會更好。

未來展望：

照度、濕度感測電路能將其輸出值作有效擷取，再由 ADC 轉為數位信號去作為 CPLD 的輸入信號，便能作到自動化溫室控制的功能；對目前氣候的異常、物種日益減少、人與自然關係的衝突，提供一個能培育植物，繁衍生長的理想環境；這也是我對科技始終抱有的信念：科技與自然界是可共存共容的，也是我撰寫此論文之原意。

(七) 參考文獻

- [1]L. A. Zadeh, "Fussy sets," Informat. Contr., Vol. 8, pp. 338-353, 1965.
- [2]H. J. Zimmermann, Fuzzy Set Theory and Its Applications. New York: Kluwer Academic, 1985.
- [3]H. J. Zimmermann, Fuzzy Sets, Decision Making, and Expert Systems. New York: Kluwer

Academic, 1987.

- [4]C. Y. Huang, C. Y. Chen, B. D. Liu, “Current-Mode Linguistic Hedge Circuit for Adaptive Fuzzy Logic Controllers,” *Electronics Letters*, Vol. 31, No.17. pp.1517-1519. 1995.
 - [5]C. C. Lee, “Fuzzy logic in control systems: fuzzy logic controller –Part I, II,” *IEEE Trans. Syst., Man, Cybern.*, Vol. 20, No. 2, pp. 404-432, Mar./Apr. 1990.
 - [6]L. A. Zadeh. “The concept of a linguistic variable and its application to approximate reasoning,” *Memorandum ERL-M 411 Berkely*, October 1973.
 - [7]E. H. Mamdani, “Application of fuzzy logic to approximate reasoning using linguistic synthesis,” *IEEE Trans Computers*, Vol. C-26, pp. 1182-1191, 1977.
 - [8]P. M. Larsen, “Industrial applications of fuzzy logic control,” *Int. J. Man, Mach, Studies*, Vol. 12, No. 1, pp. 3- 10, 1980.
 - [9]Y. Tsukamoto, “An approach to fuzzy reasoning method,” *Fuzzy Set Theory and Applications*, M. M. Gupta, R. K. Ragade, and R. R. Yager, Eds. Amsterdam: North-Holland, 1979.
 - [10]T. Takagi, and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Trans. Syst. Man. Cybern.*, Vol. SMC-15, No. 1, pp. 116-132, 1985.
 - [11] 鐘國家,“感測器原理與應用實習”,全華出版社.1996.
 - [12] 魏境鴻,“CPLD 設計與應用”,高長文化.2000.
 - [13] 林容益,“CPLD 與數位邏輯設計應用”,知行文化.2001.
- 作者簡介:賴高獅 高雄高工資訊科技佐/義守大學電子工程學系碩士班進修中.