



# MIAT系統設計與硬體高階合成方法論

## —以指紋辨識晶片開發為例

義守大學電機系

機器智慧與自動化技術(MIAT)實驗室

陳慶瀚

[pierre@isu.edu.tw](mailto:pierre@isu.edu.tw)

2002年9月30日



# 大綱

PART 1：系統設計與硬體高階合成方法論

PART 2：方法論的實作

PART 3：「指紋辨識系統單晶片」開發實例



# PART 1

## 系統設計與硬體高階合成方法論



# 問題

由於產品的生命週期越來越短，而功能的多樣性和複雜度則不斷增加，現有的資訊/電子產品研發工作常需大幅仰賴系統工程師的個別設計經驗，因此使得產品開發的設計時間、成本、可靠度和重覆使用性都難以掌握，而且團隊的研發也難以在嚴謹的、演繹的基礎上進行分工、協同的設計。

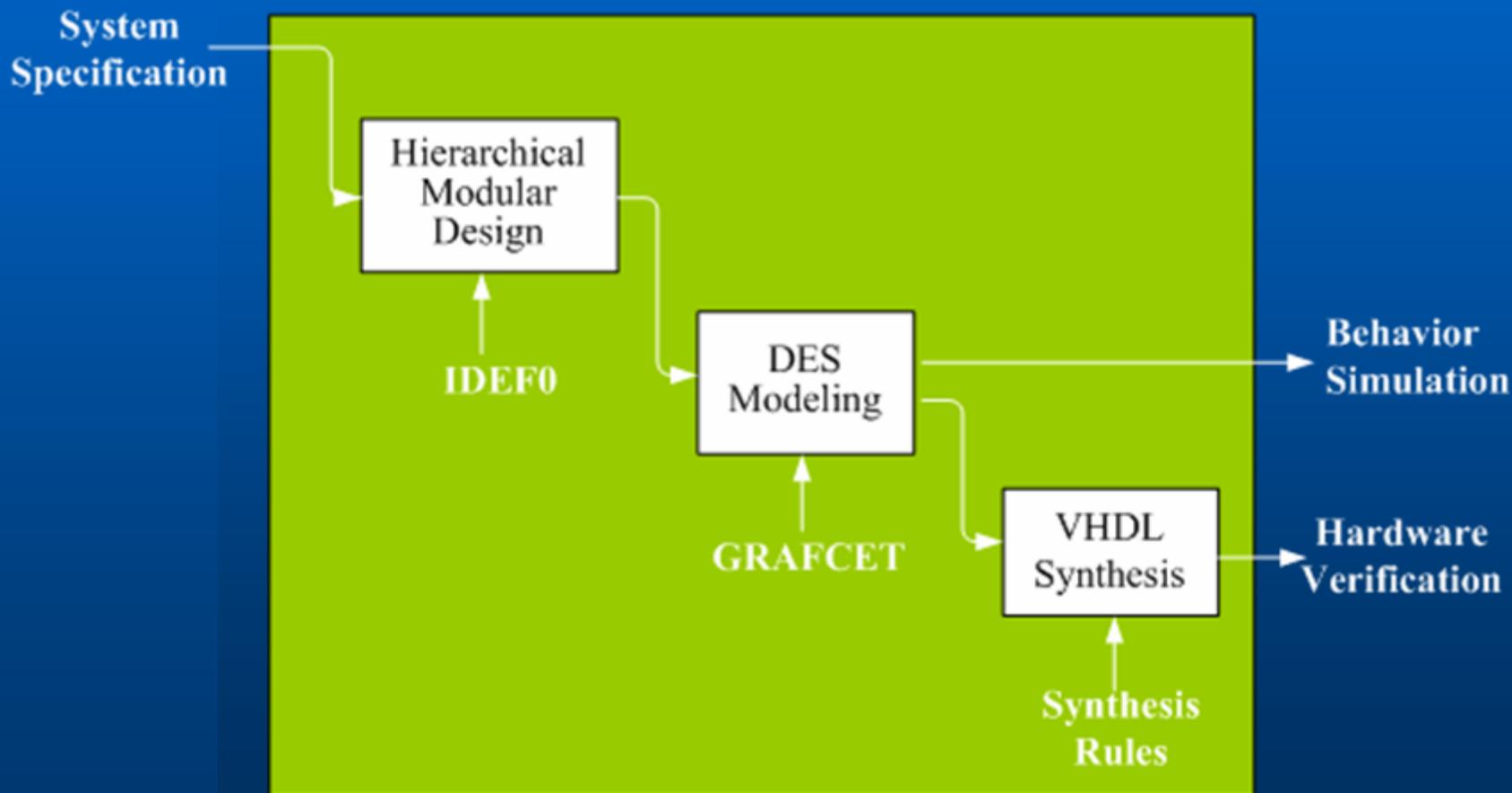


# 目標

規劃一個系統設計和高階合成的方法論 (methodology)，將不同的技術、工具和方法，整合在一個階層式、模組化的設計流程，以便達到系統自動化設計和快速原型化(Rapid Prototyping)之目的。



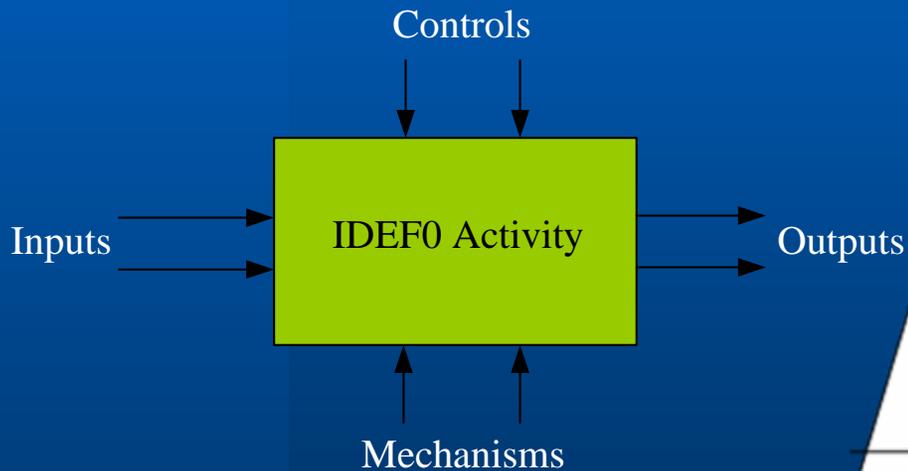
# 系統設計與高階合成方法論



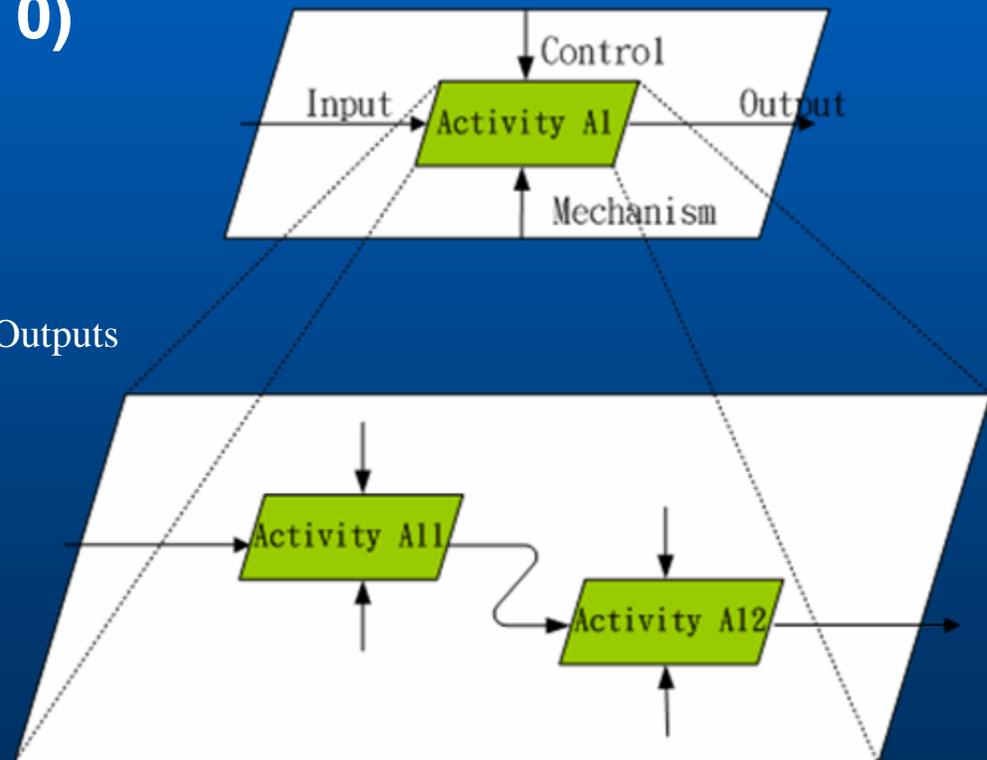


# 階層式與模組化設計

- IDEF0(Integrated Computer-Aided Manufacturing (ICAM) DEFinition 0)



IDEF0基本模組功能方塊與箭頭意義

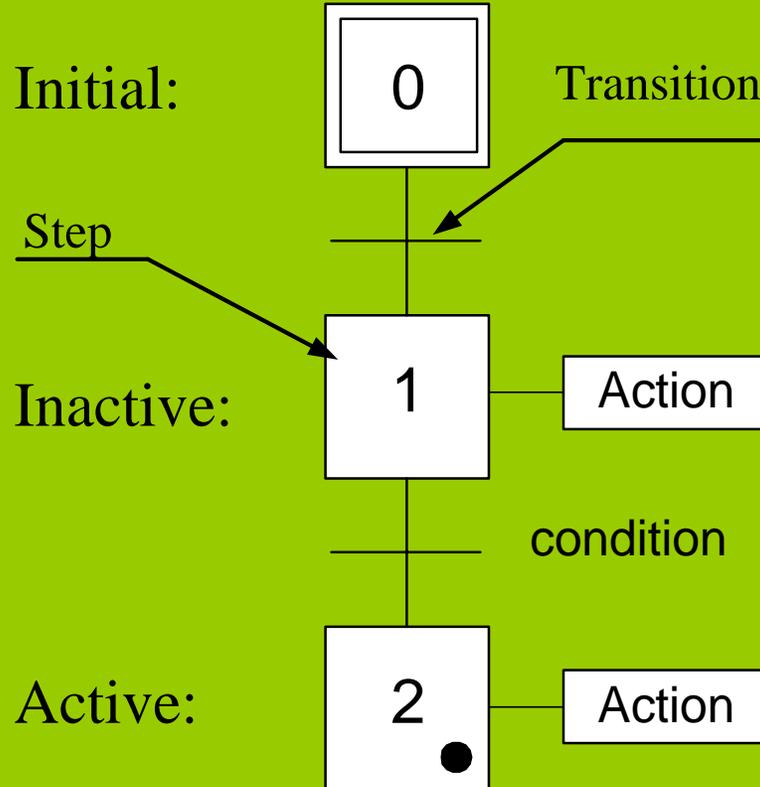


IDEF0 階層化架構



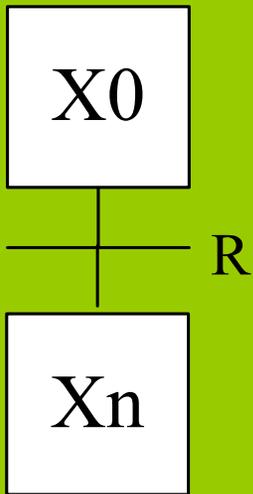
# 離散事件(DES)建模

## GRAFCET

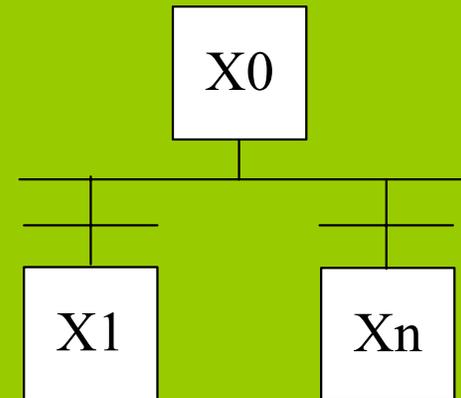
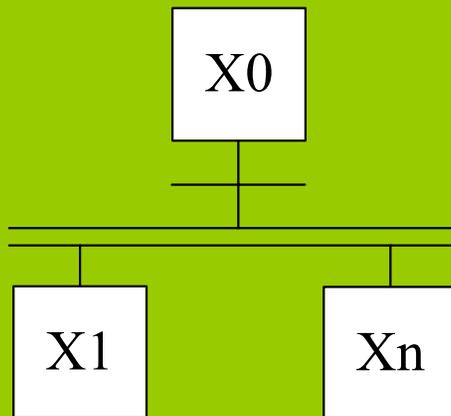
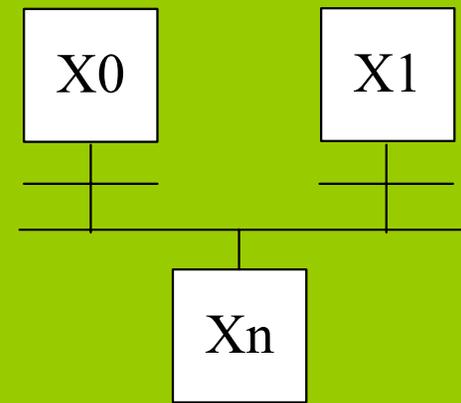
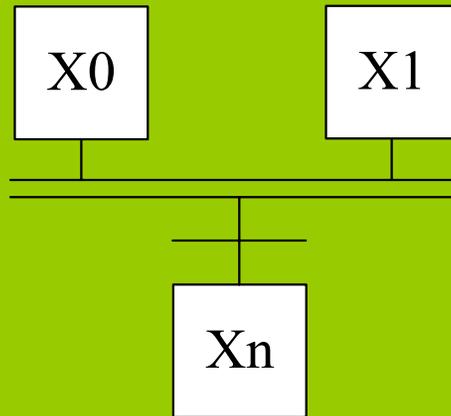




# GRAFCET結構

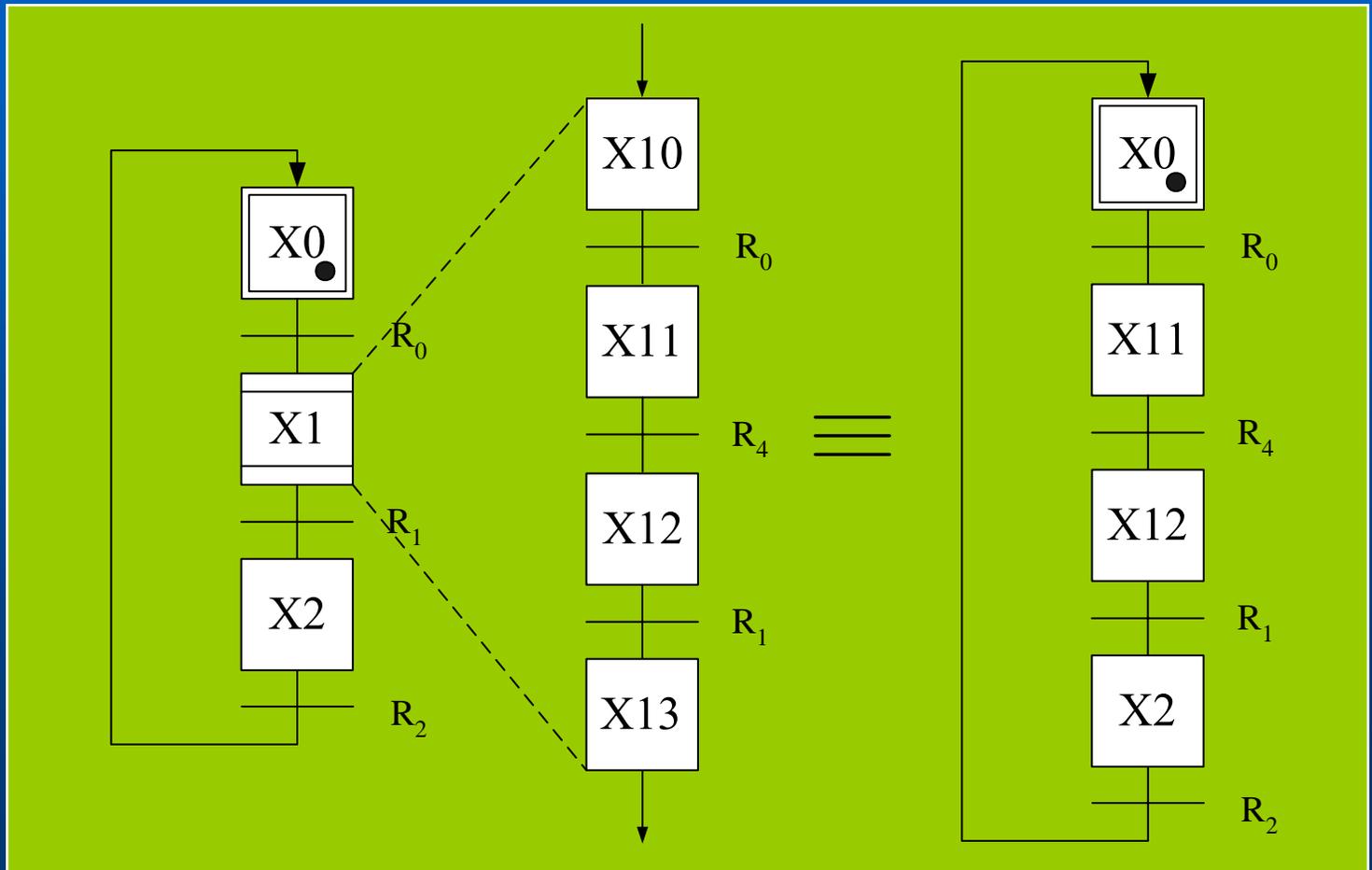


Grafcet  
建構單元



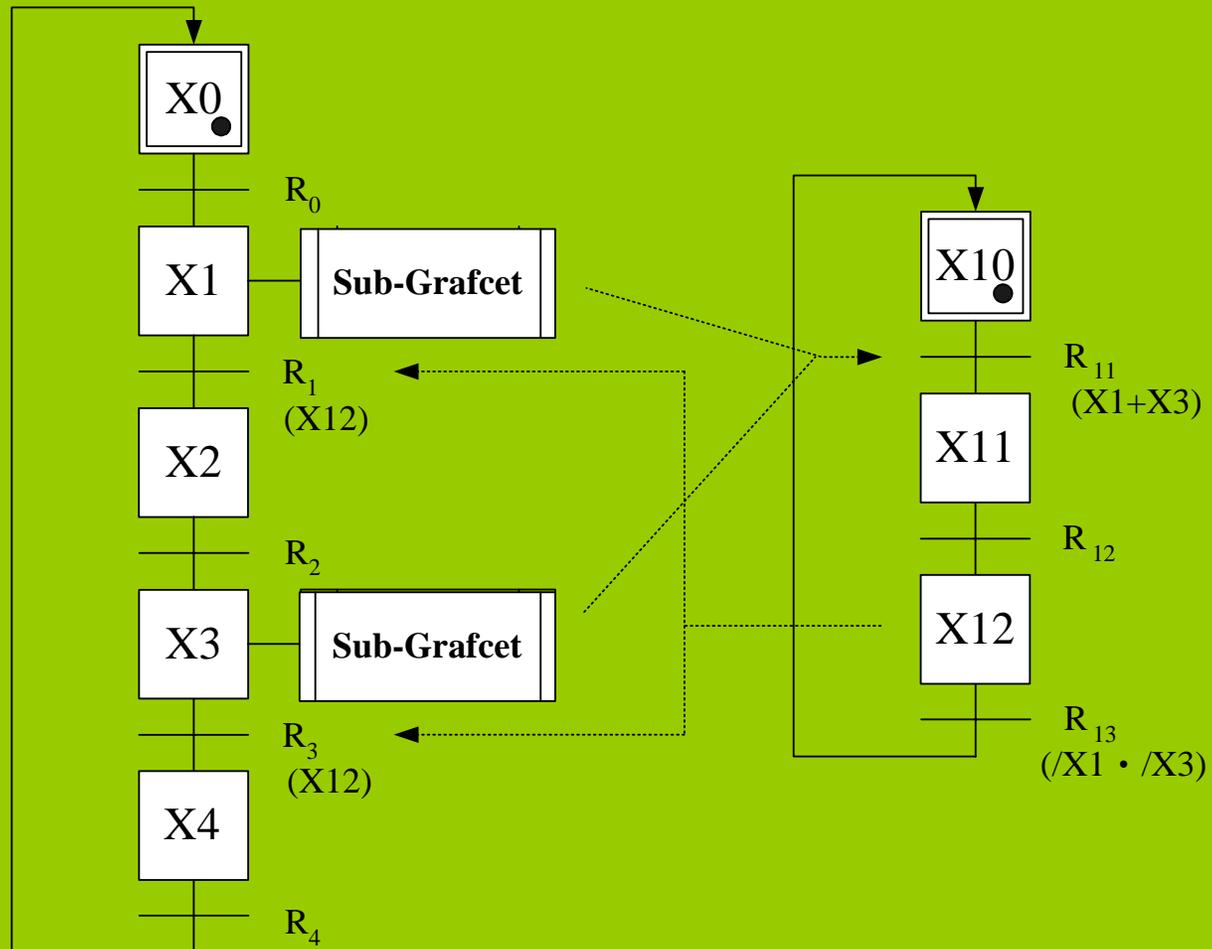


# Macrostep



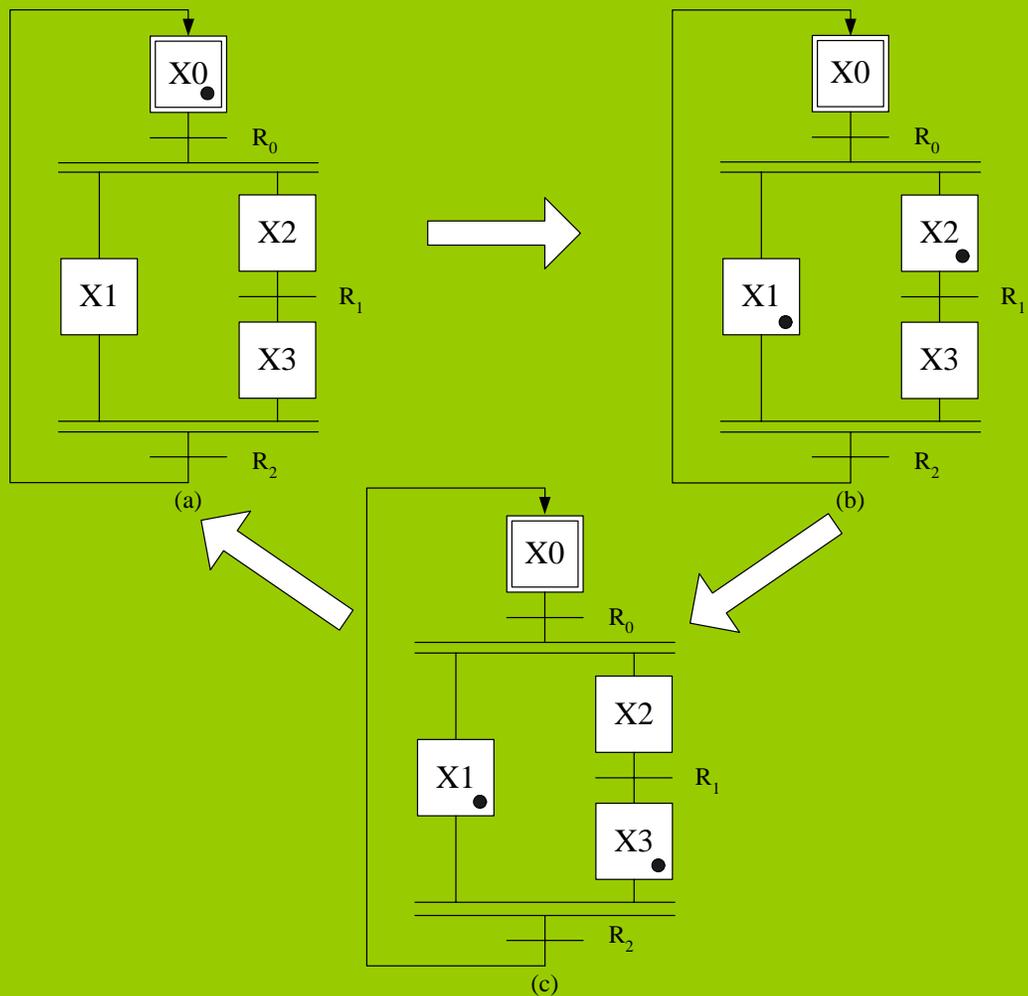


# Sub-Grafcet



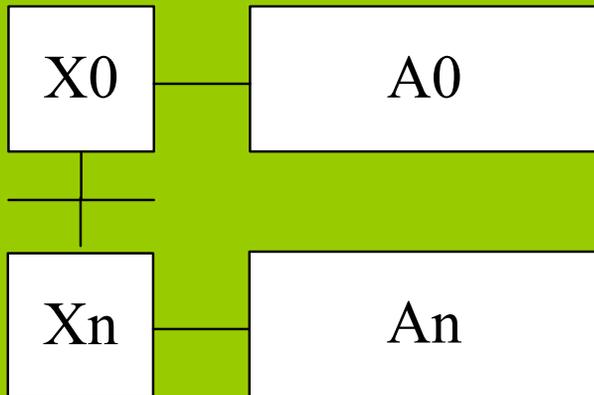


# Grafcet的離散事件模擬





# VHDL 電路合成



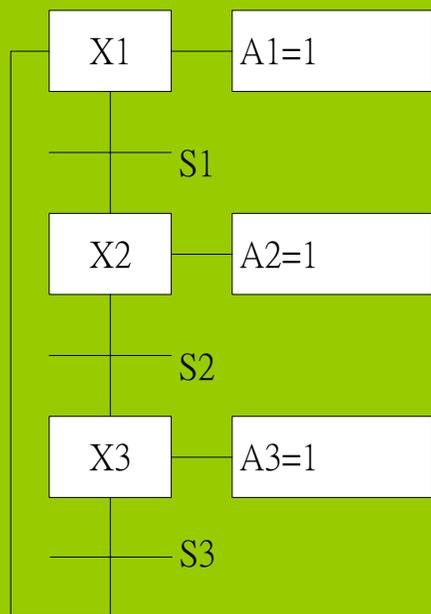
if X0='1' and R0='1' then X0<='0'; Xn<='1';  
end if;

A0 <= X0 ;

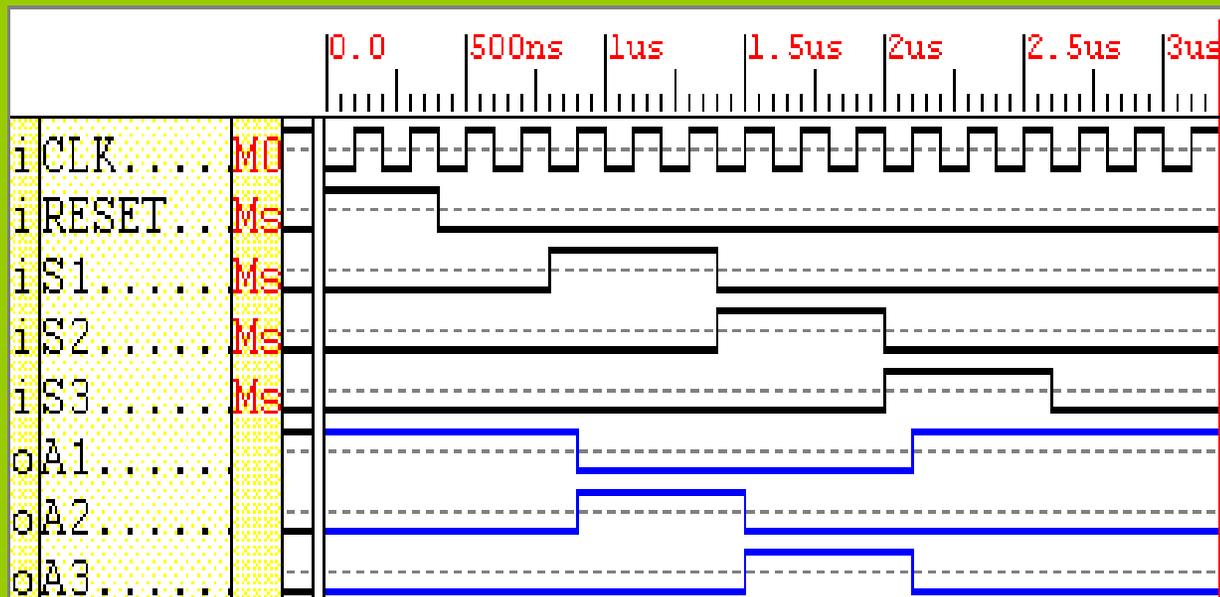
An <= Xn ;



# DES合成實例—功能模擬



(a)



(b)



# DES合成實例—VHDL合成

```
architecture DEMO1_arch of DEMO1 is
    signal X1,X2,X3:STD_LOGIC;
begin
    process(CLK)
        begin
            if RESET='1' then

                X1<='1';
                X2<='0';
                x3<='0';

            elsif CLK'EVENT and CLK='1' then

                if X1='1' and S1='1' then X1<='0';X2<='1'; end if;
                if X2='1' and S2='1' then X2<='0';X3<='1'; end if;
                if X3='1' and S3='1' then X3<='0';X1<='1'; end if;

            end if;

            A1<=X1;
            A2<=x2;
            A3<=X3;

        end process;
    end DEMO1_arch;
```



# PART 2

## 方法論實作



# 方法論實作流程

方法·  
演算法

C/Java  
功能  
驗證

函式  
模組  
切割

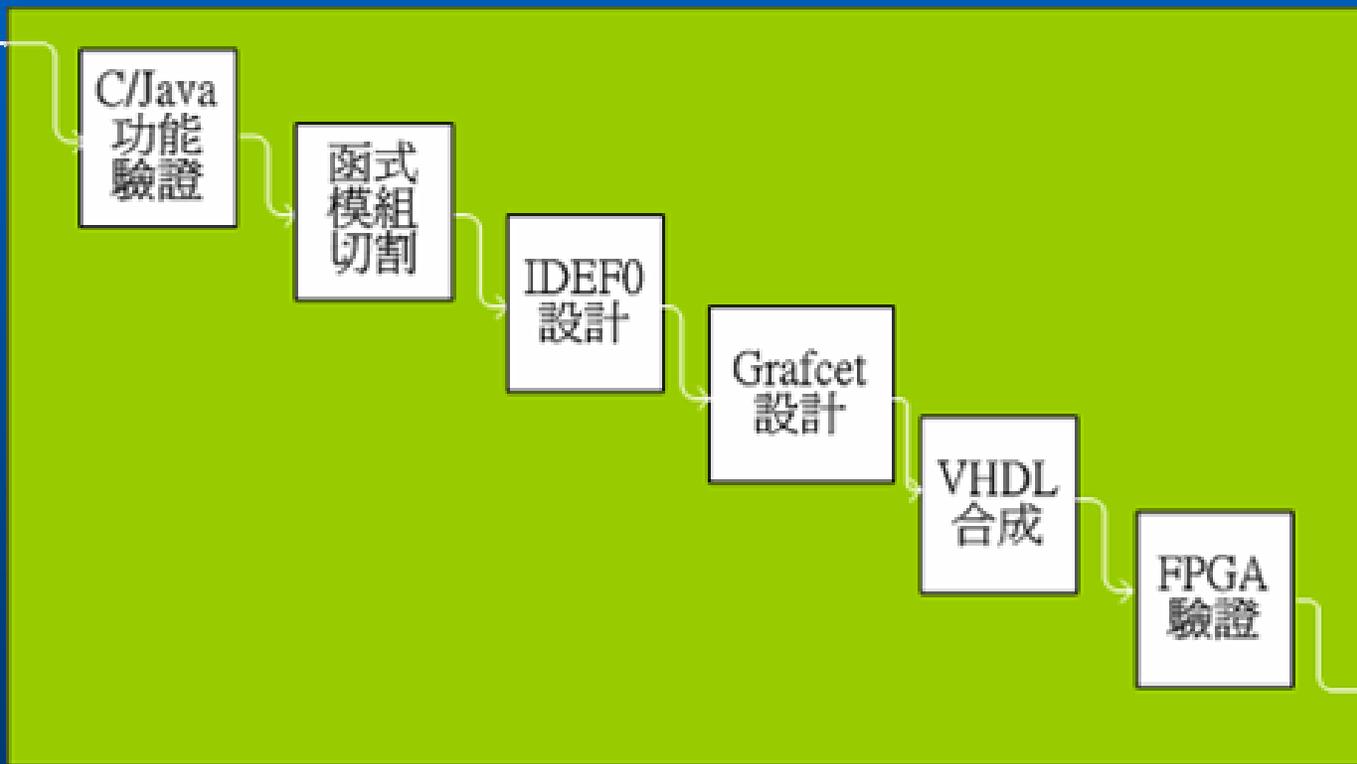
IDEFO  
設計

Grafcet  
設計

VHDL  
合成

FPGA  
驗證

硬體  
實現





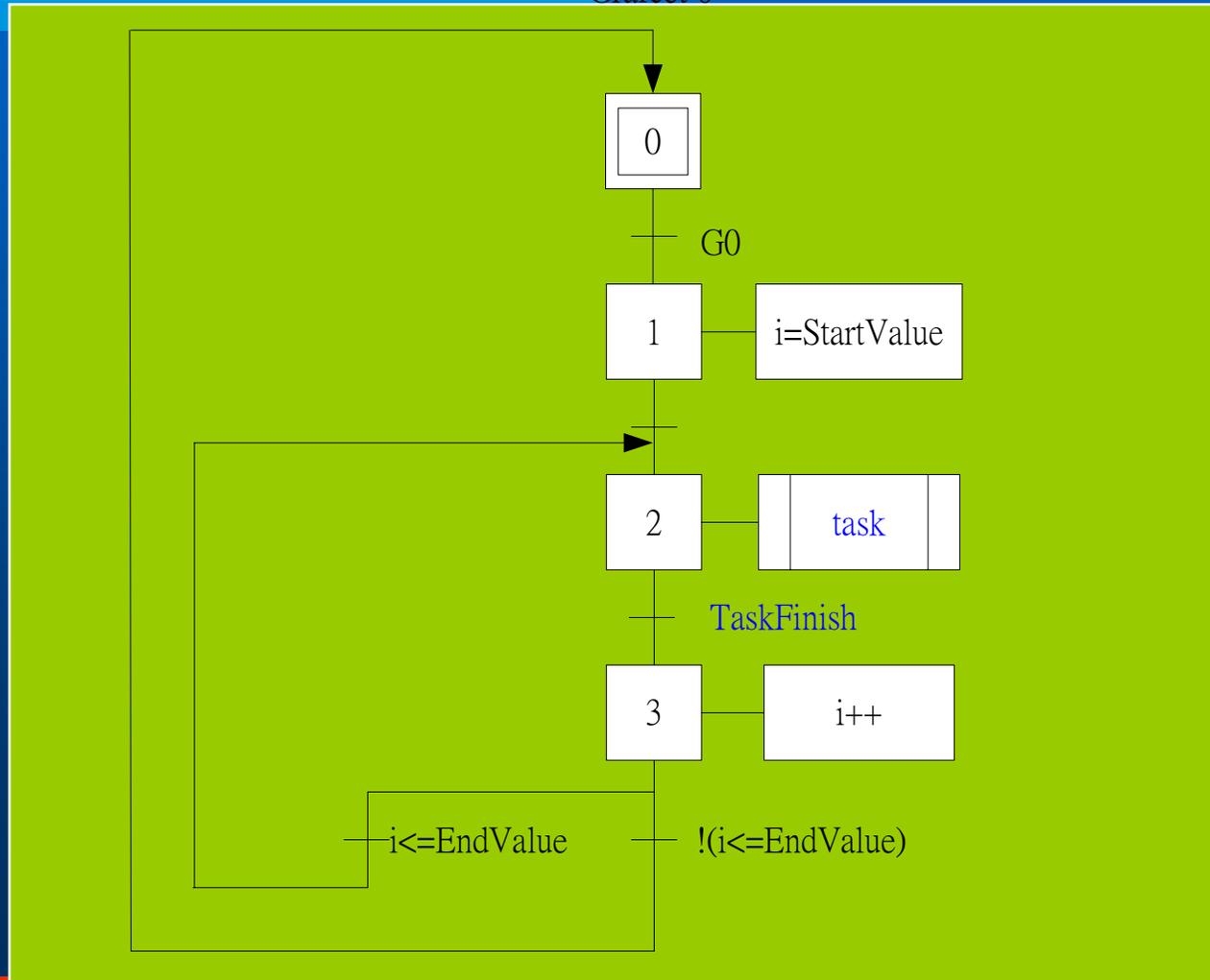
# 高階演算法—FOR迴圈

```
for (i=Start_Value; i<=EndValue; i++)  
{  
    task ;  
}
```



# 演算法的Grafcet建模

Grafcet 0



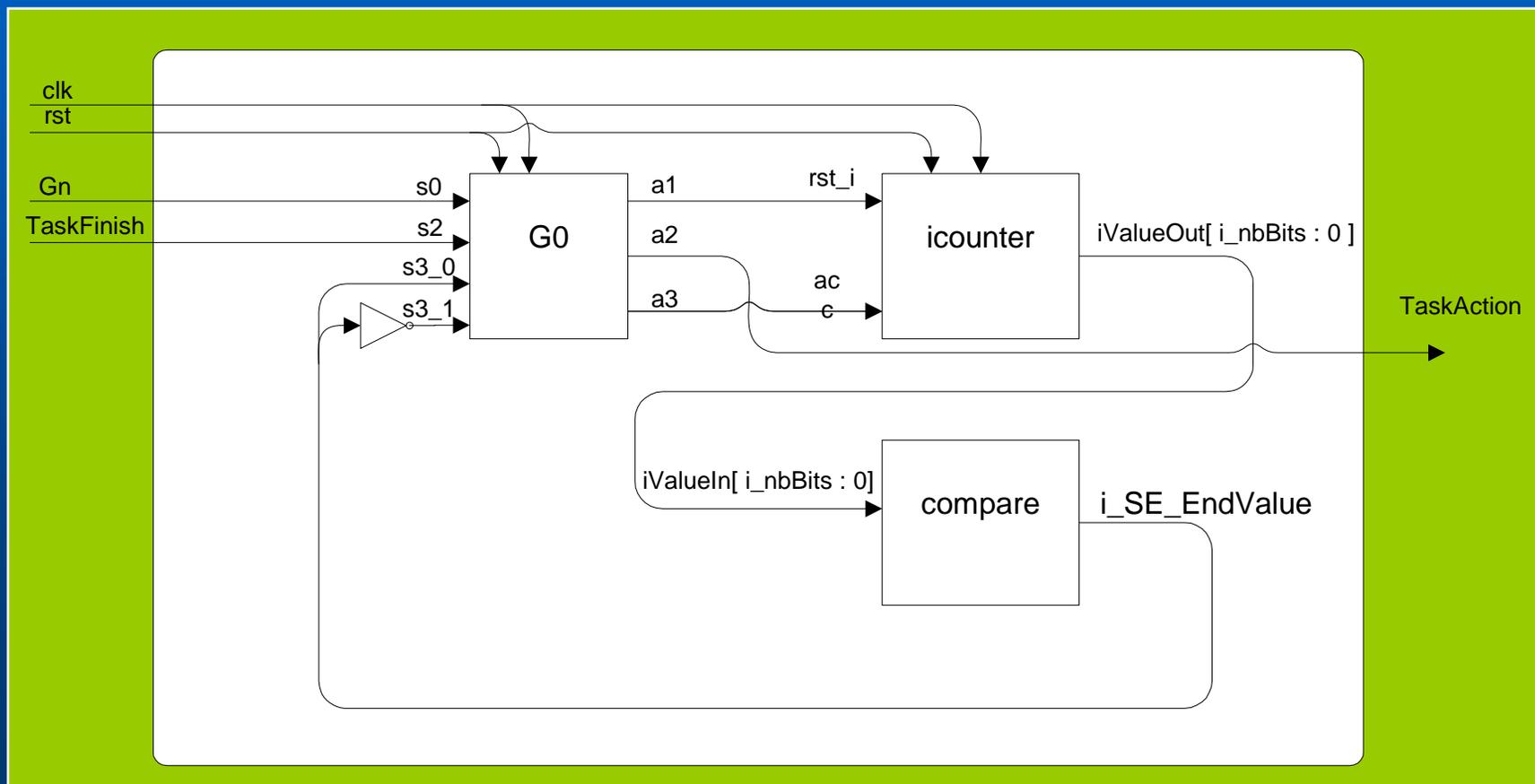


# Grafcet的VHDL合成

```
-----  
g0 :BLOCK  
    signal x0,x1,x2,x3 : std_logic;  
begin  
    process(clk,rst)  
    begin  
        if rst='0' then  
            x0<='1';  
            x1<='0';  
            x2<='0';  
            x3<='0';  
        elsif clk'event and clk='1' then  
            if x0='1' and s0='1' then x0<='0'; x1<='1';  
            elsif x1='1' then x1<='0'; x2<='1';  
            elsif x2='1' and s2='1' then x2<='0'; x3<='1';  
            elsif x3='1' and s3_0='1' then x3<='0'; x2<='1';  
            elsif x3='1' and s3_1='1' then x3<='0'; x0<='1';  
            end if;  
        end if;  
    end process;  
  
    a1<=x1;  
    a2<=x2;  
    a3<=x3;  
  
END BLOCK g0;  
-----
```

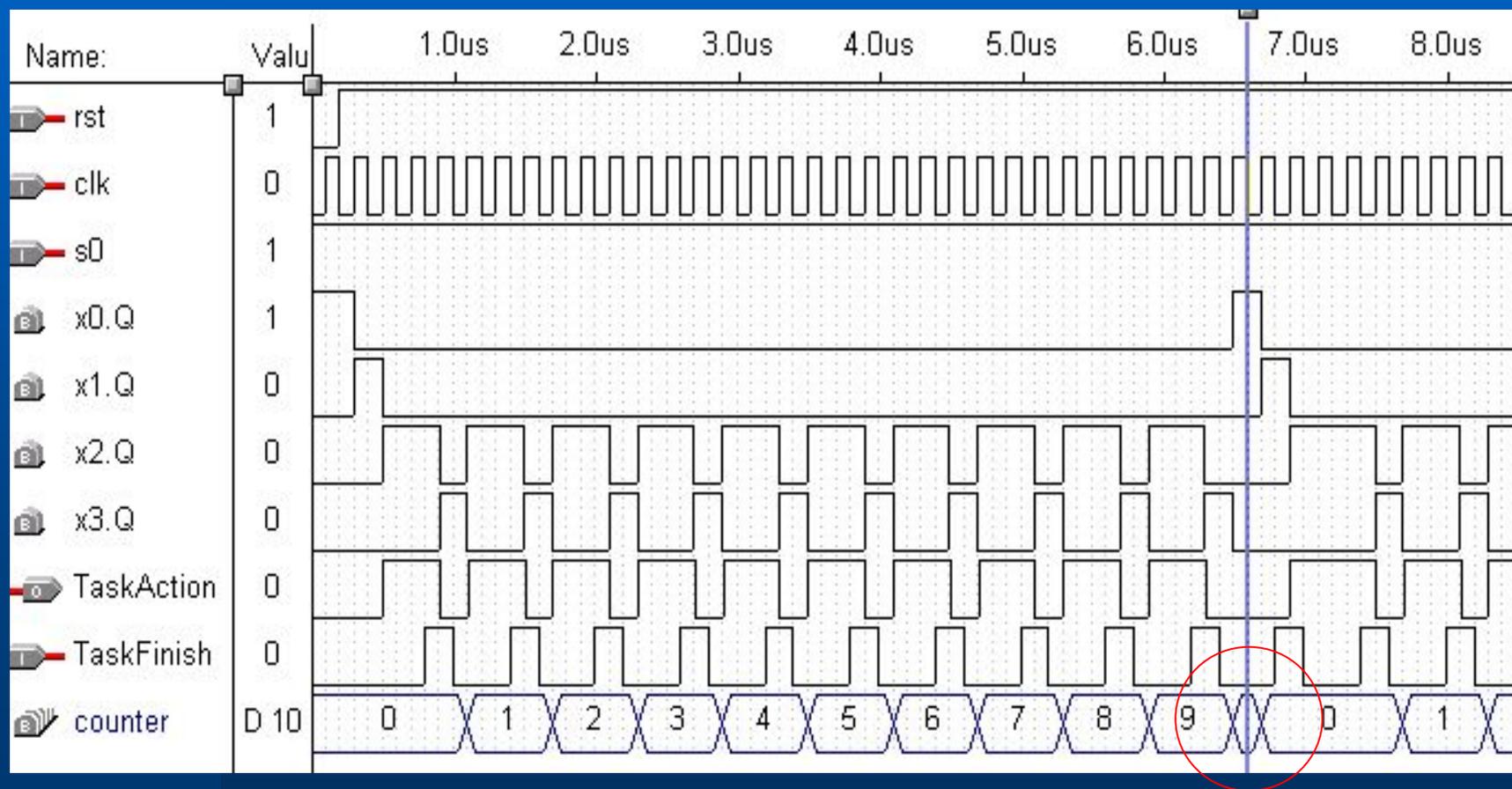


# 電路方塊圖





# 功能驗證





# 應用範例：資料排序電路設計

```
unsigned char ram[10]={1,3,5,7,9,2,4,6,8,0};
unsigned char temp1;
unsigned char temp2;

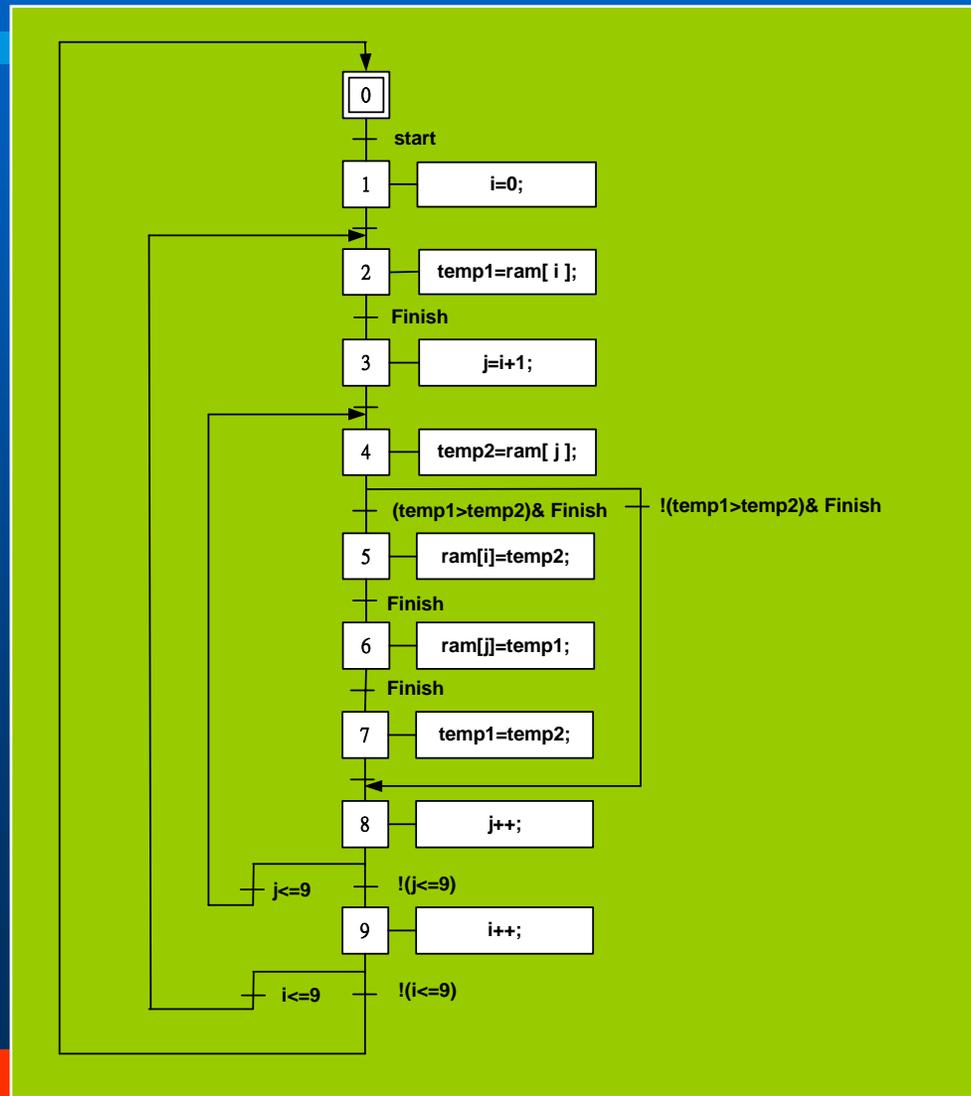
for(int i=0;i<10;i++)
{
    temp1=ram[i];
    for(int j=i+1;j<10;j++)
    {
        temp2=ram[j];
        if(temp1>temp2)
        {
            ram[i]=temp2;
            ram[j]=temp1;
            temp1=temp2;
        }
    }
}
```

D:\A\_WORK200203\HYBRID CONTR

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

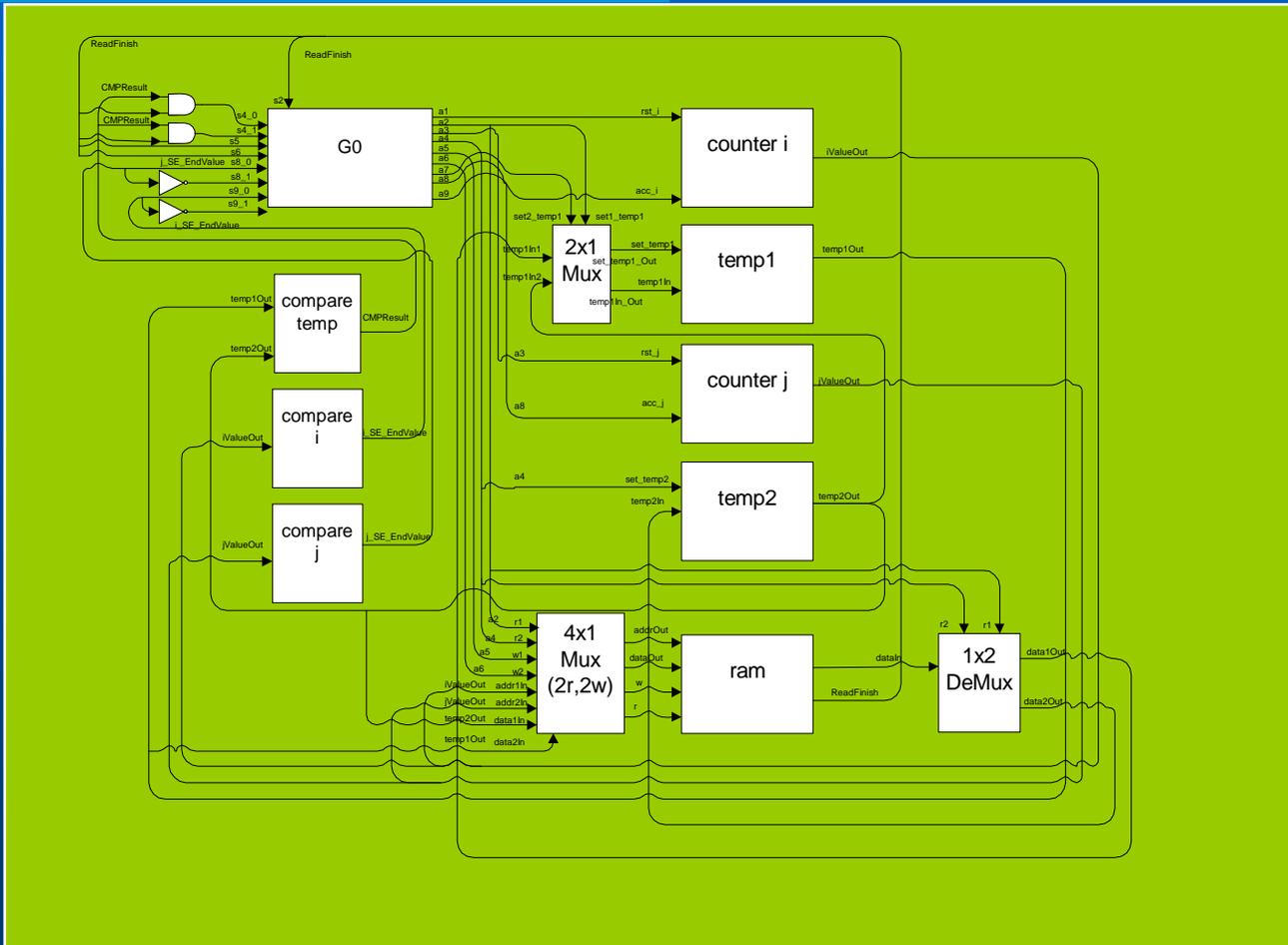


# 應用範例：演算法建模



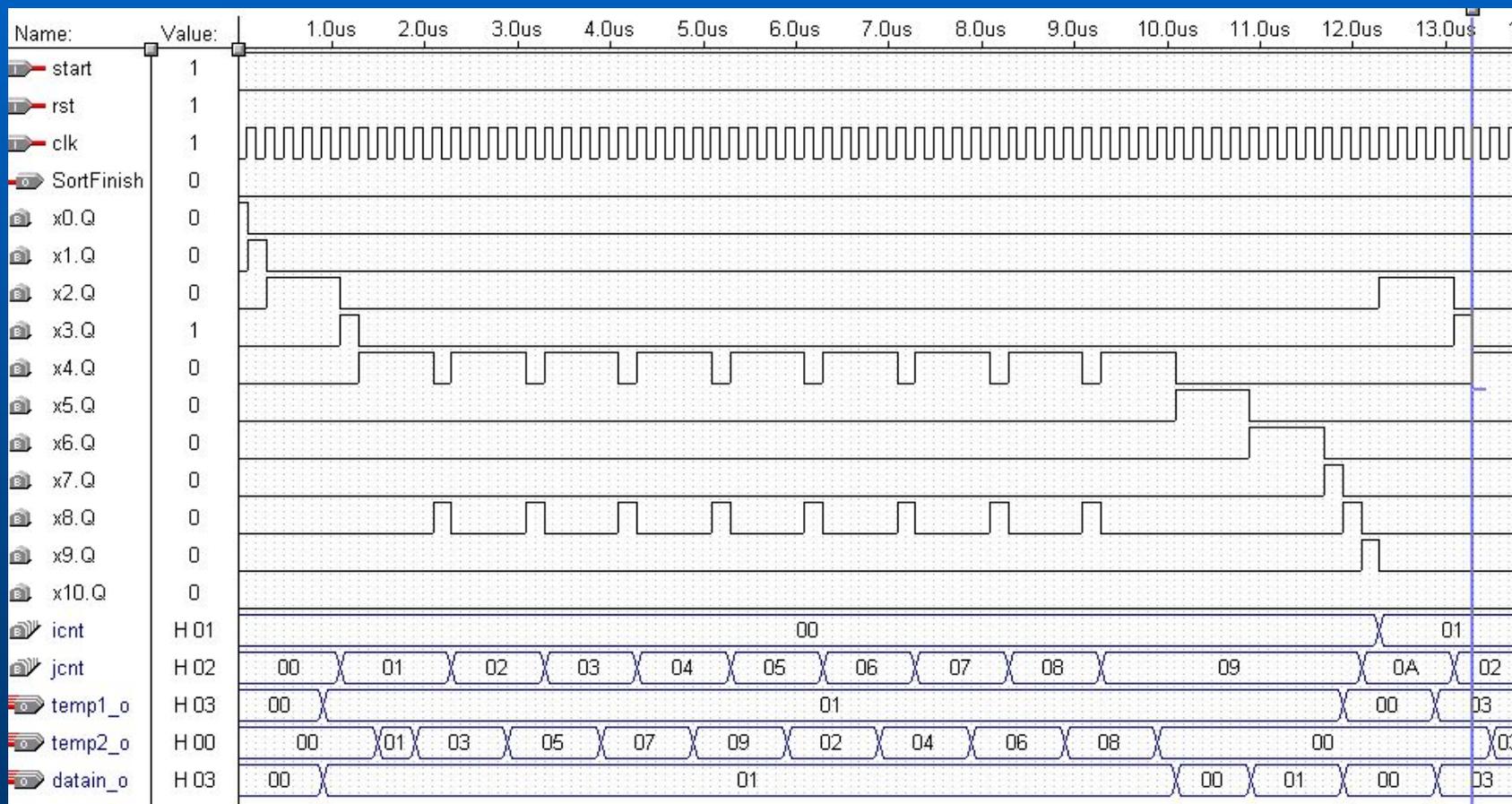


# 應用範例：資料排序電路





# 應用範例：功能模擬







# PART 3

方法論應用

— 「指紋辨識系統單晶片」開發實例

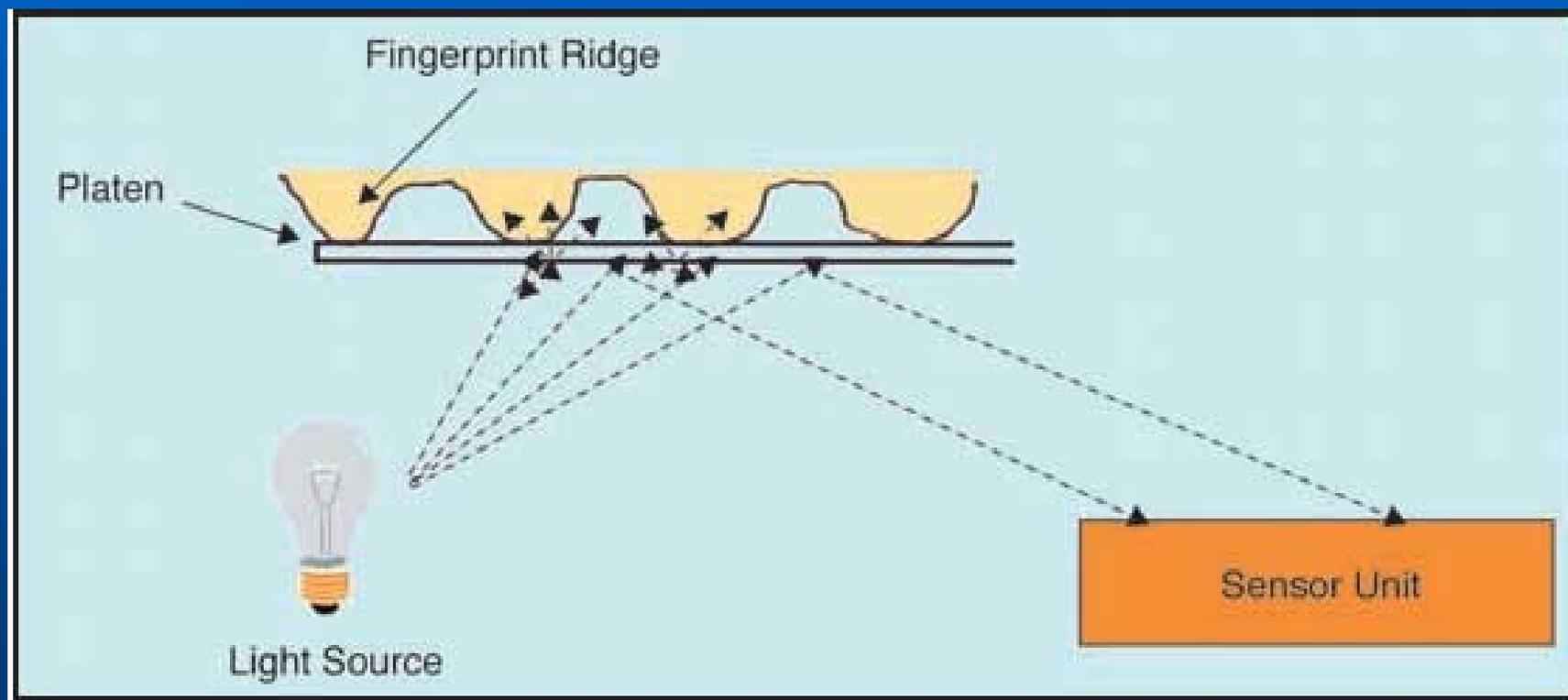


# 指紋辨識方法

1. 取像(Image Acquisition)
2. 影像處理(Image Processing)
3. 指紋特徵點抽取(Minutia Detection)
4. 特徵點比對(Point Matching)

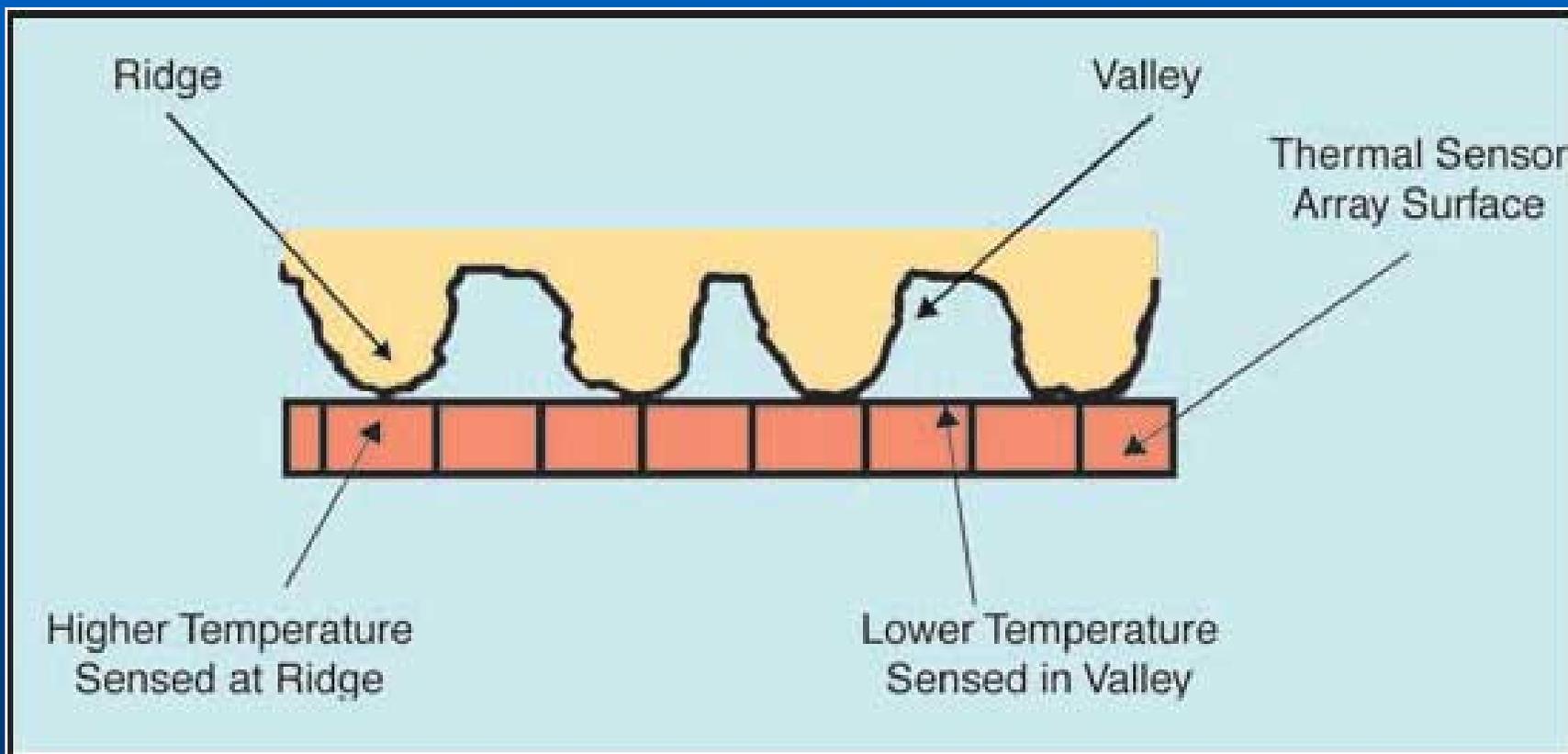


# 指紋取像—光學式



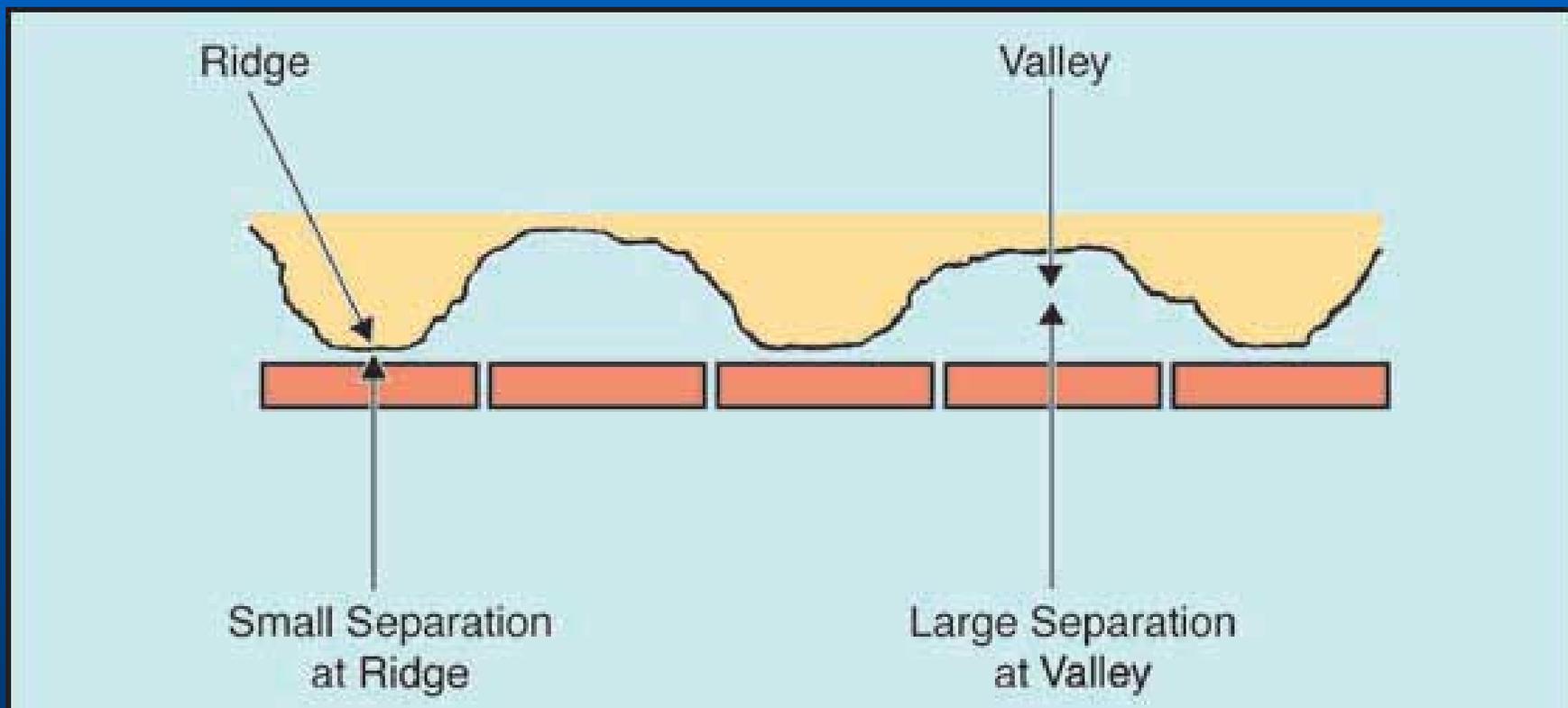


# 指紋取像—熱感式



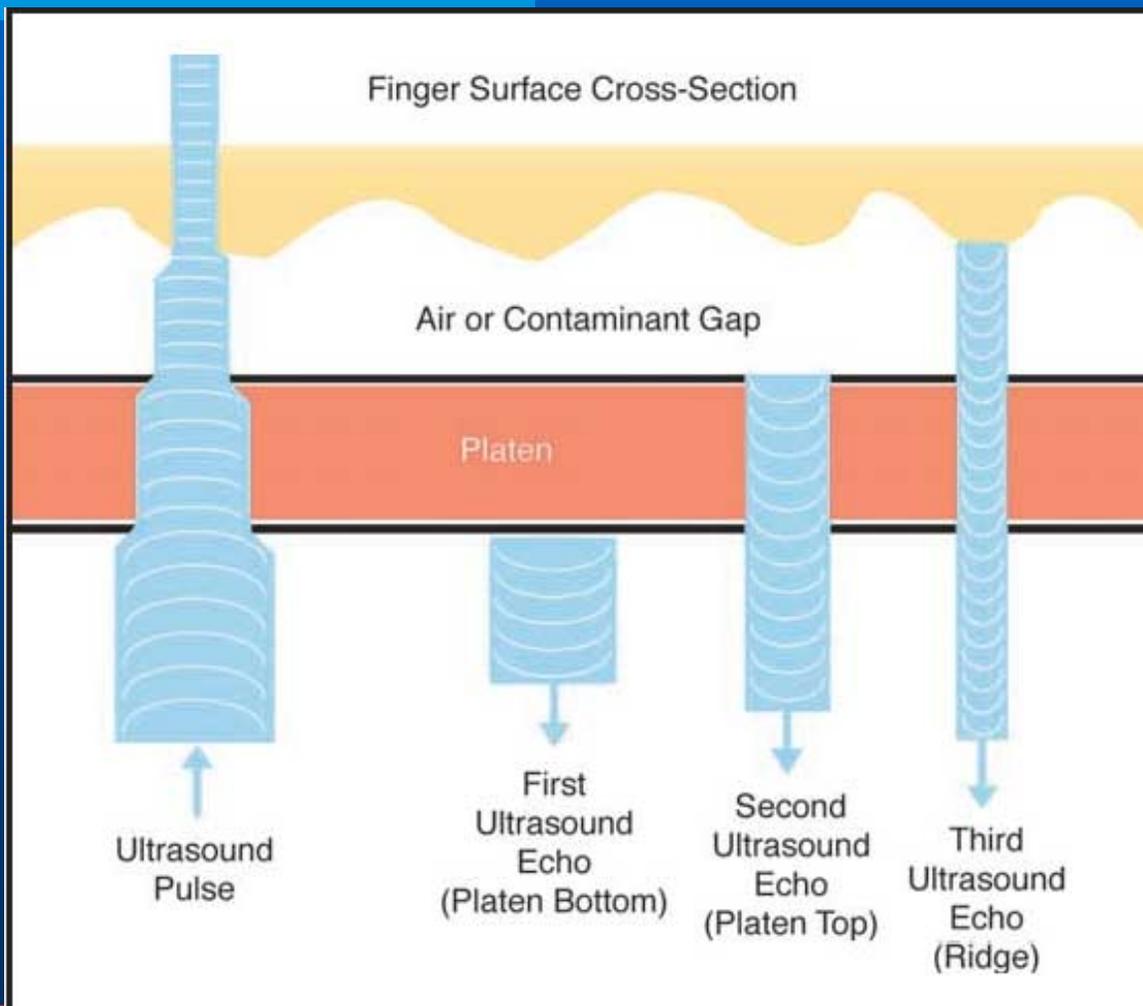


# 指紋取像一電容式



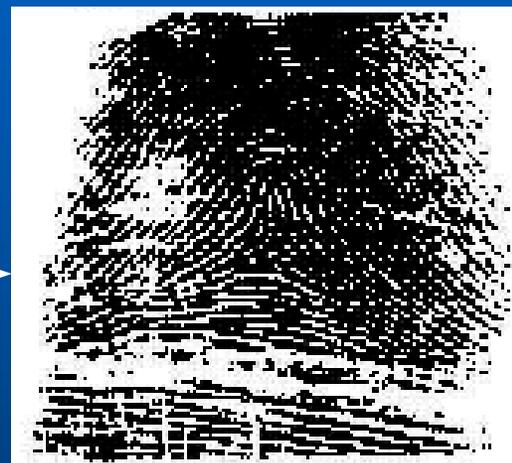
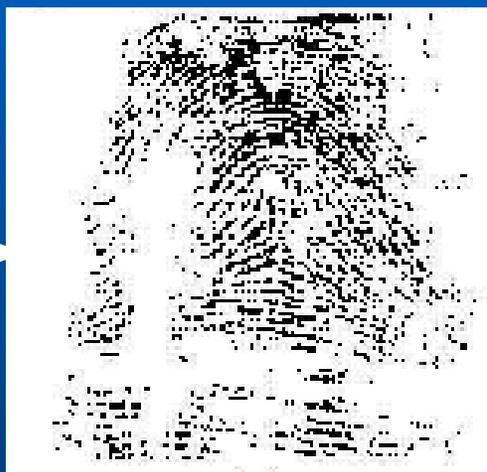
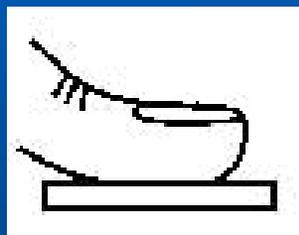


# 指紋取像—超音波





# 影像處理(一)



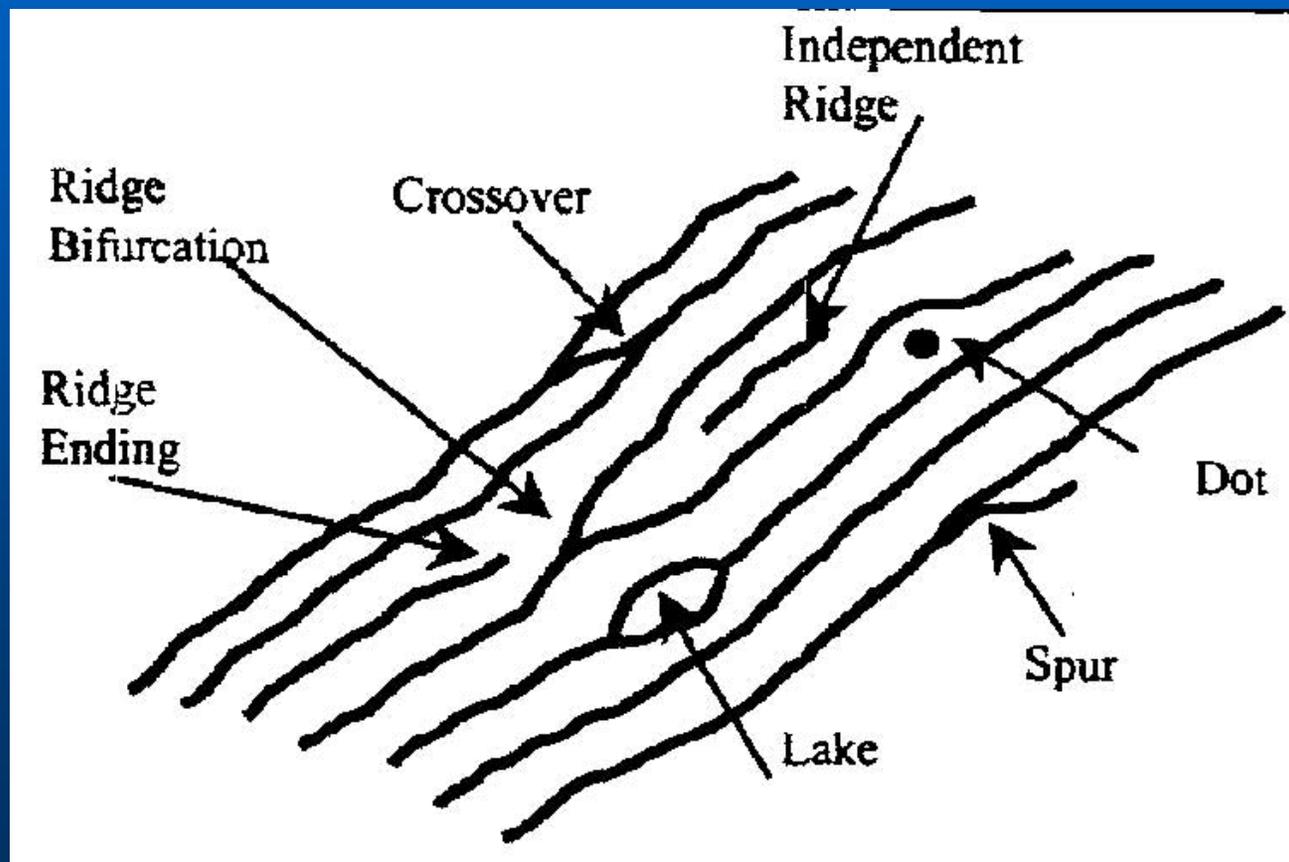


# 影像處理(二)





# 指紋特徵點(Minutia)



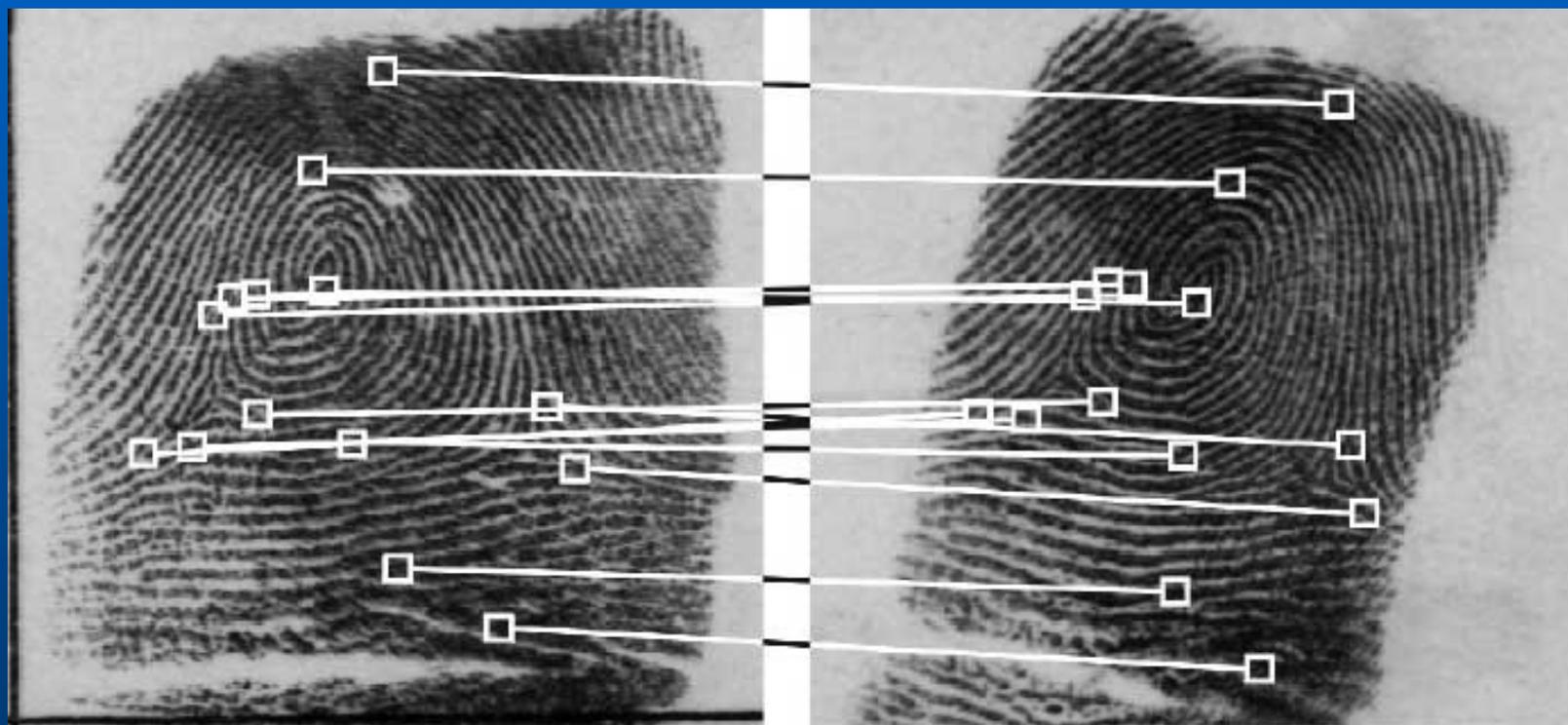


# 指紋特徵點抽取





# 特徵點比對





# 傳統指紋辨識方法的缺點

- 演算法複雜度高
- 辨識時間長
- 雜訊、感測器的適應性不佳
- 實現成本高



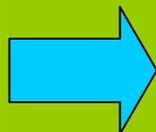
# MIAT指紋辨識方法

- 無須解析式數學模型
- 具有旋轉不變的影像特徵化
- 免除影像前處理
- 強大的適應能力
- 擁有泛用途的建模機制

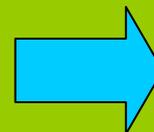


# Rotation-Invariant 紋路描述

|   |   |   |
|---|---|---|
| a | b | c |
| h |   | d |
| g | f | e |



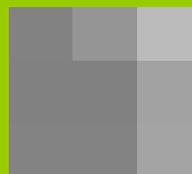
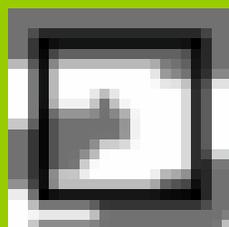
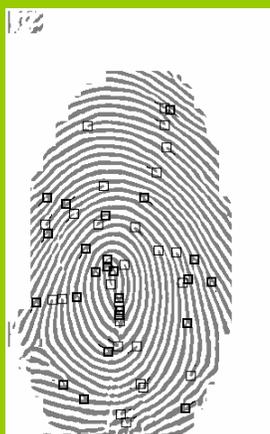
Circular  
Auto-  
Correlation



$$X_p = \begin{bmatrix} \tilde{R}_{xx}(0) \\ \tilde{R}_{xx}(1) \\ \tilde{R}_{xx}(2) \\ \tilde{R}_{xx}(3) \\ \tilde{R}_{xx}(4) \\ \tilde{R}_{xx}(5) \\ \tilde{R}_{xx}(6) \\ \tilde{R}_{xx}(7) \end{bmatrix}$$



# Circular Auto-Correlation



|   |   |   |
|---|---|---|
| a | b | c |
| h |   | d |
| g | f | e |

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| a | b | c | d | e | f | g | h |
|---|---|---|---|---|---|---|---|

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| a | b | c | d | e | f | g | h |
|---|---|---|---|---|---|---|---|

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| a | b | c | d | e | f | g | h |
|---|---|---|---|---|---|---|---|

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| h | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|

⋮

$$\tilde{R}_{XX}(0) = a \times a + b \times b + \dots + h \times h$$

$$\tilde{R}_{XX}(1) = a \times h + b \times a + \dots + h \times g$$

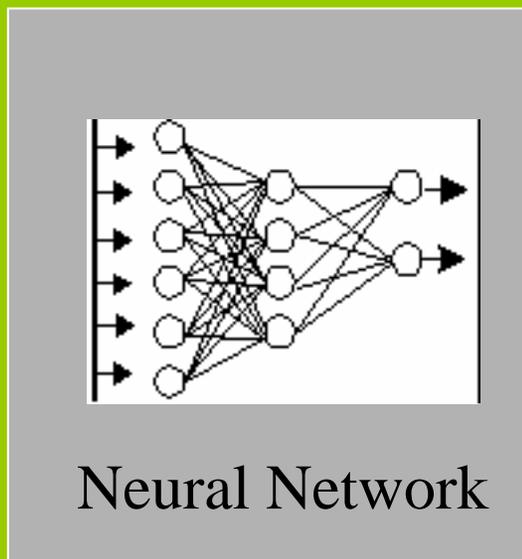
⋮

$$\tilde{R}_{XX}(7)$$



# 神經網路指紋特徵點偵測

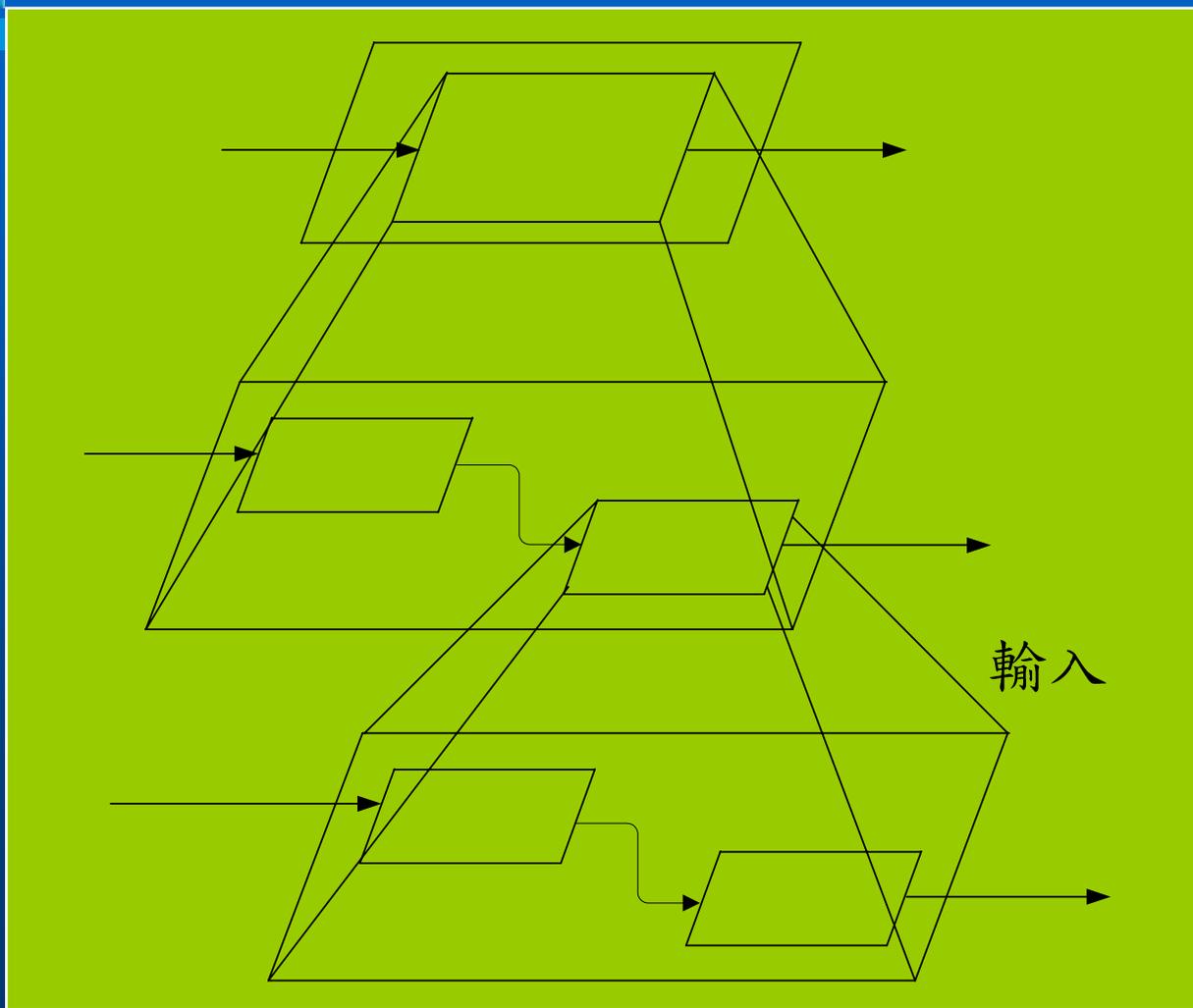
$$X_p = \begin{bmatrix} \tilde{R}_{xx}(0) \\ \tilde{R}_{xx}(1) \\ \tilde{R}_{xx}(2) \\ \tilde{R}_{xx}(3) \\ \tilde{R}_{xx}(4) \\ \tilde{R}_{xx}(5) \\ \tilde{R}_{xx}(6) \\ \tilde{R}_{xx}(7) \end{bmatrix}$$



| 1 <sup>st</sup><br>Output | Is<br>Minutia? |
|---------------------------|----------------|
| $< 0.5$                   | NO             |
| $\geq 0.5$                | YES            |



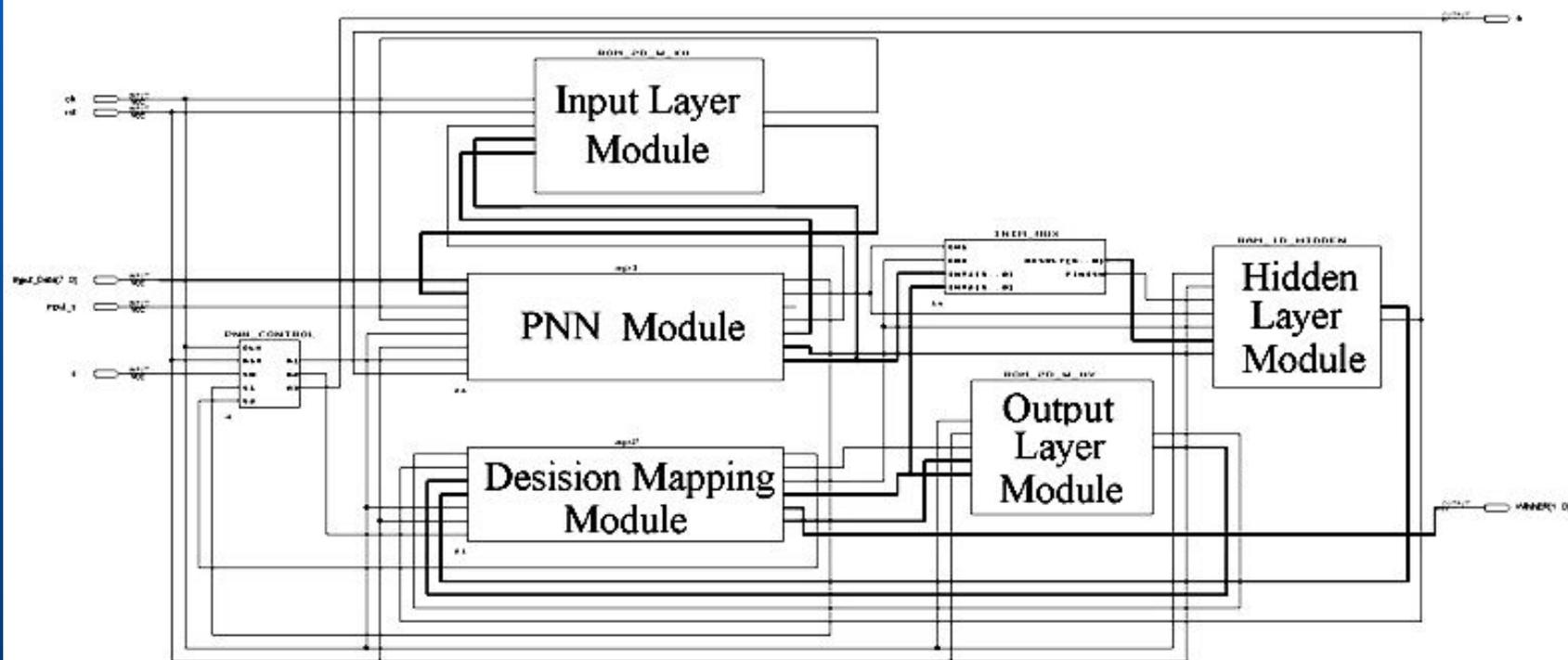
# 機率式神經網路分類器架構



機率式



# 機率式神經網路電路合成





# 粒子群最佳化演算法 (Particle Swarm Optimization)

$$v_{j,d}(t) = v_{j,d}(t-1) + \phi_{j,d}^{(1)}(t)(x_{j,d}(t^*) - x_{j,d}(t-1)) \\ + \phi_{j,d}^{(2)}(t)(x_{j,d}(t^\#) - x_{j,d}(t-1))$$

$$x_{j,d}(t) = x_{j,d}(t-1) + v_{j,d}(t)$$



# PSO與指紋特徵點比對

A  
指  
紋

B  
指  
紋

Particle  
Swarm  
Optimization

A 與 B 之最小變形量

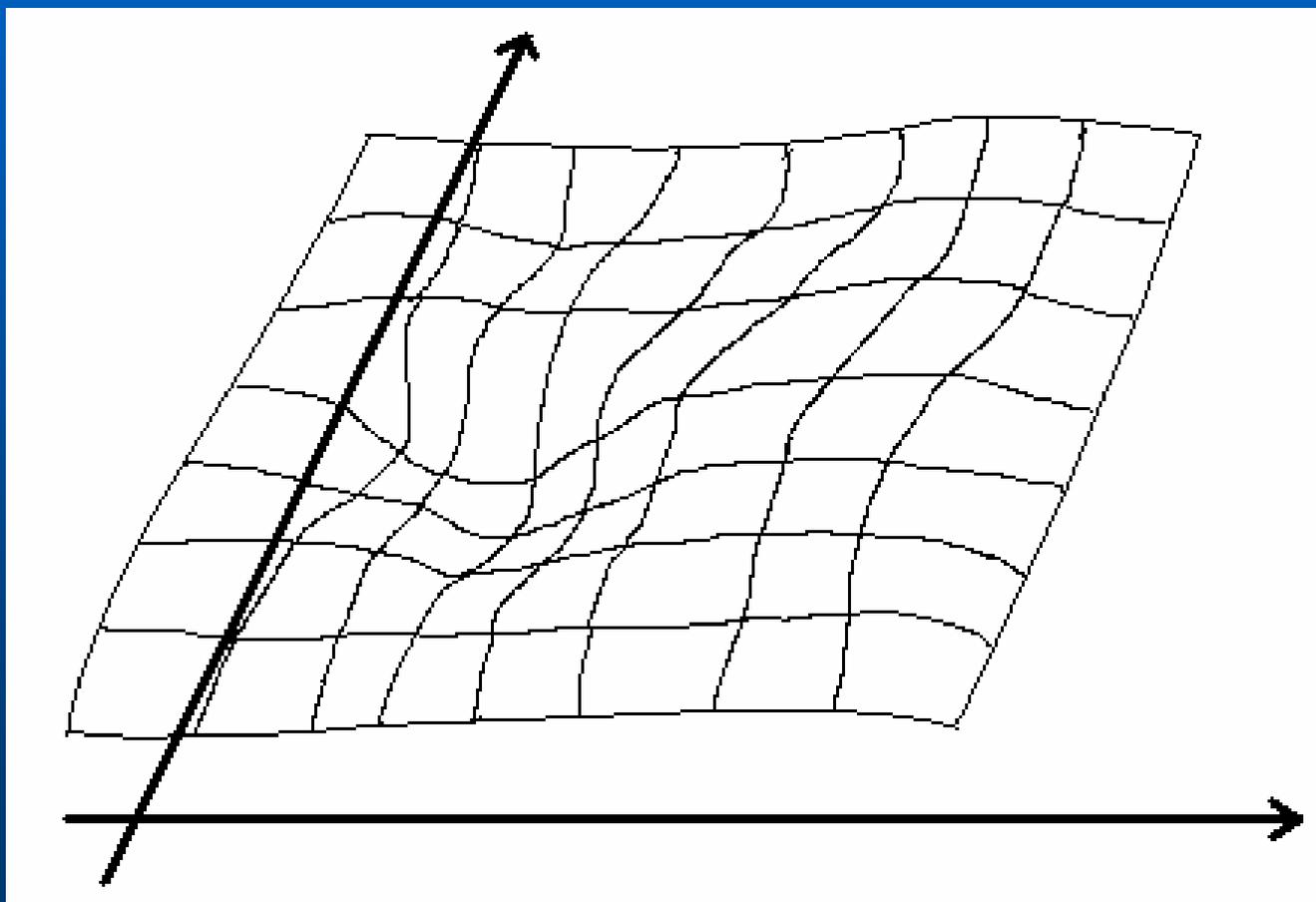
dX X 軸方向平移變形量

dY Y 軸方向平移變形量

$d\theta$  以影像左上角為  
原點之旋轉變形量

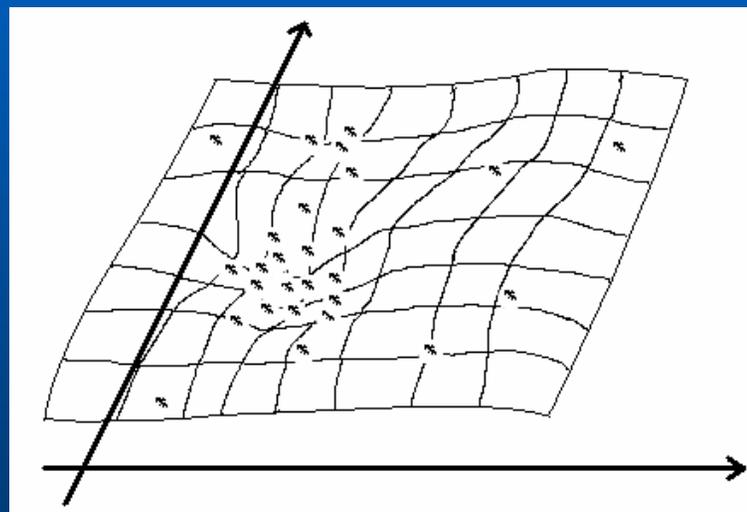
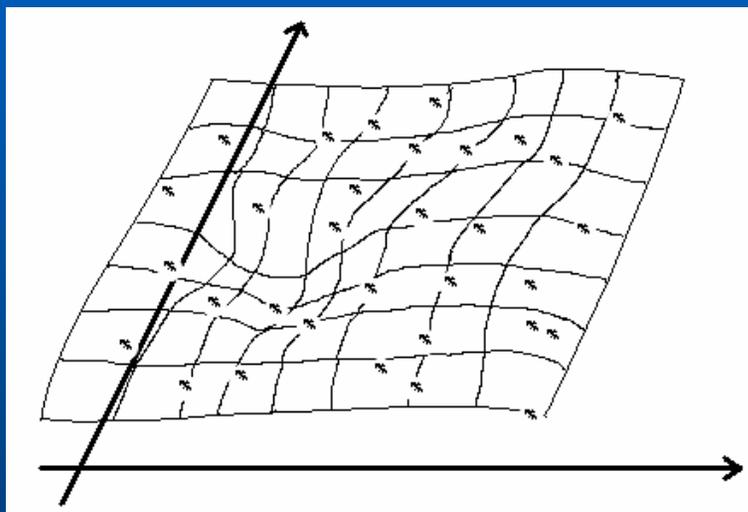


# 搜尋最佳解的指紋比對問題





# PSO搜尋最佳解

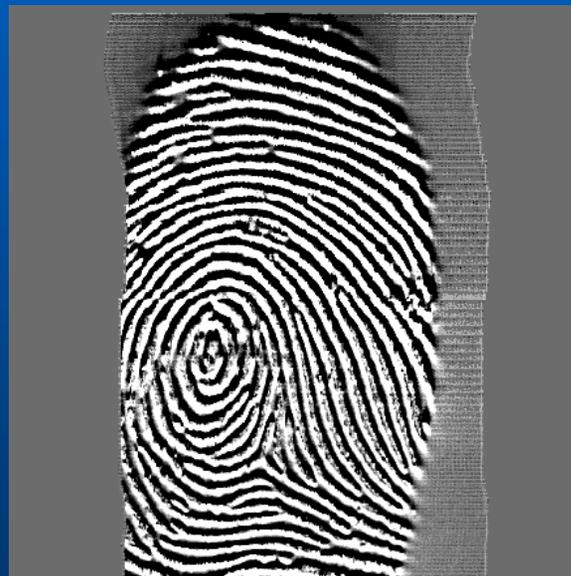
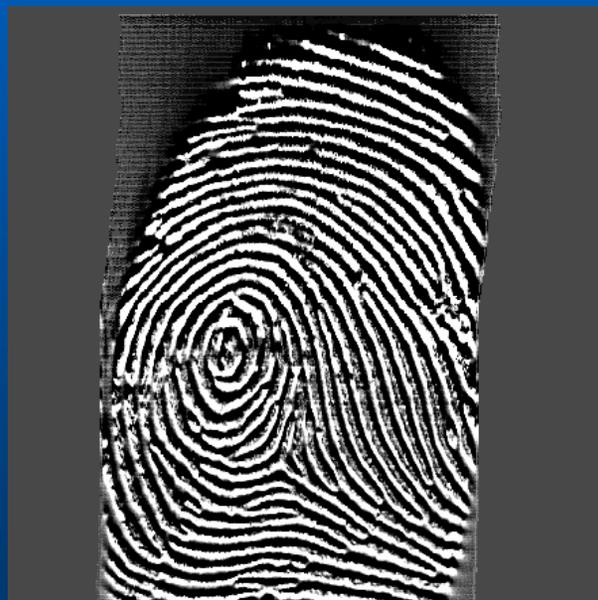


Demo



# PSO搜尋指紋比對最佳解

相同指紋

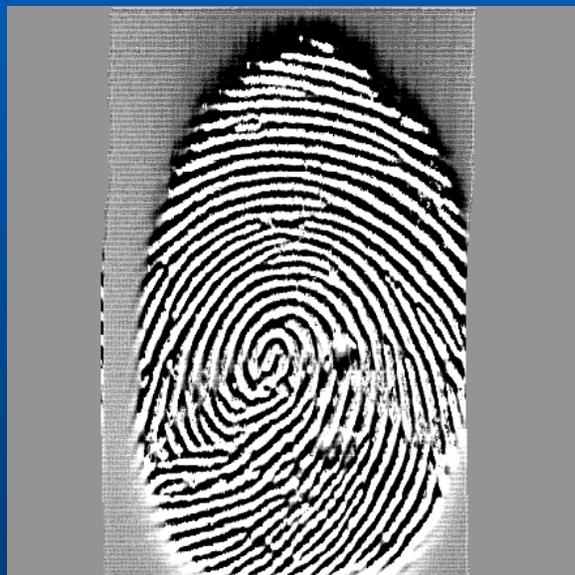


$$dX = -21.8548 \quad dY = 23.6224 \quad d\theta = 357.19 \quad E = 139.178$$



# PSO搜尋指紋比對最佳解

不同指紋



$$dX = 506.952 \quad dY = 254.741 \quad d\theta = 113.611 \quad E = 446.602$$



# 指紋辨識晶片開發

- **Software Design :**

Algorithms elaboration; C/C++ Programming ; Discretization of algorithms

- **Hardware Design :**

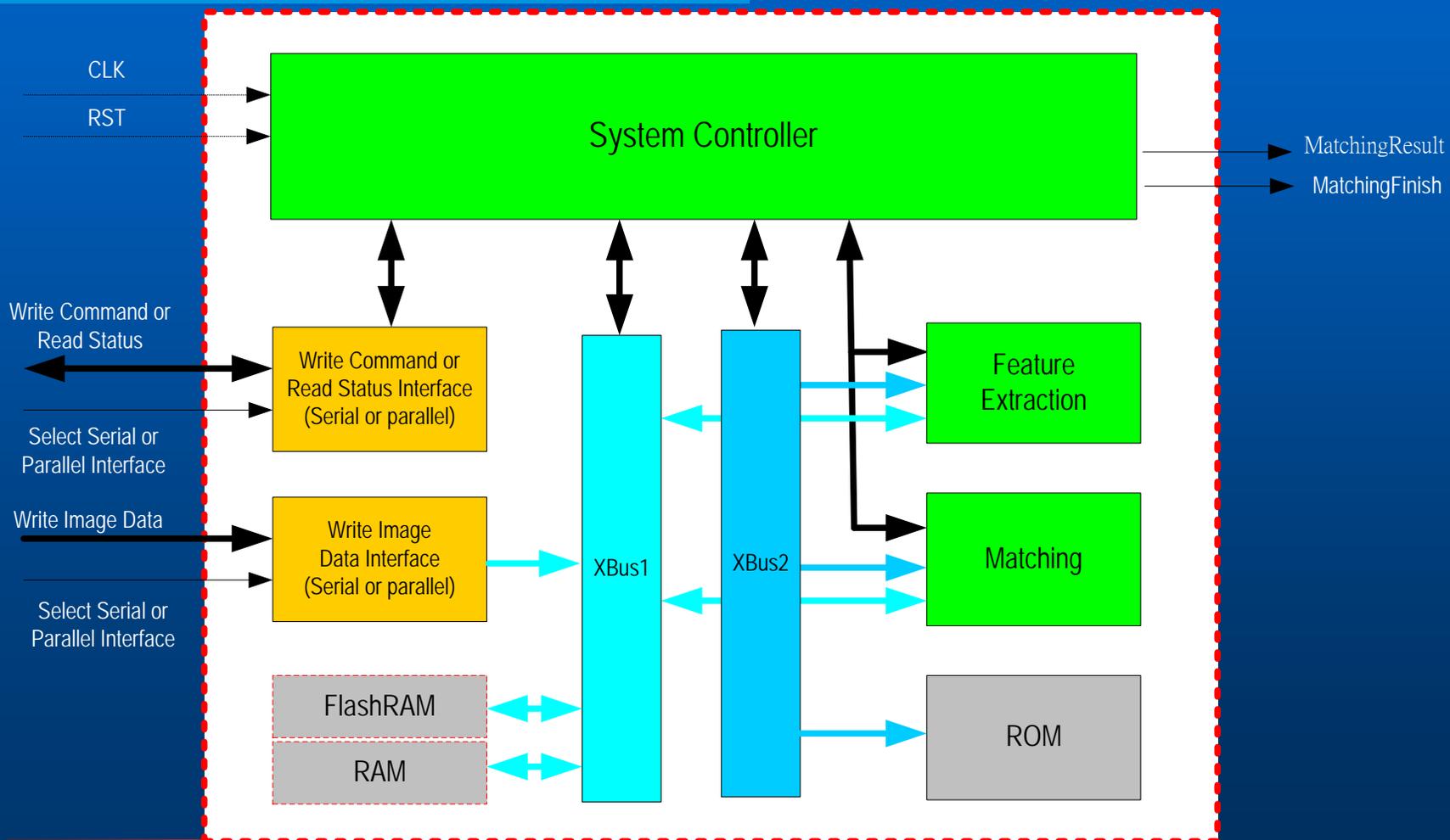
GRAFCET modeling, VHDL coding, FPGA verification

- **CAD :**

High-level synthesis, Automatic Design, Software/Hardware co-design, System pre-design/simulation

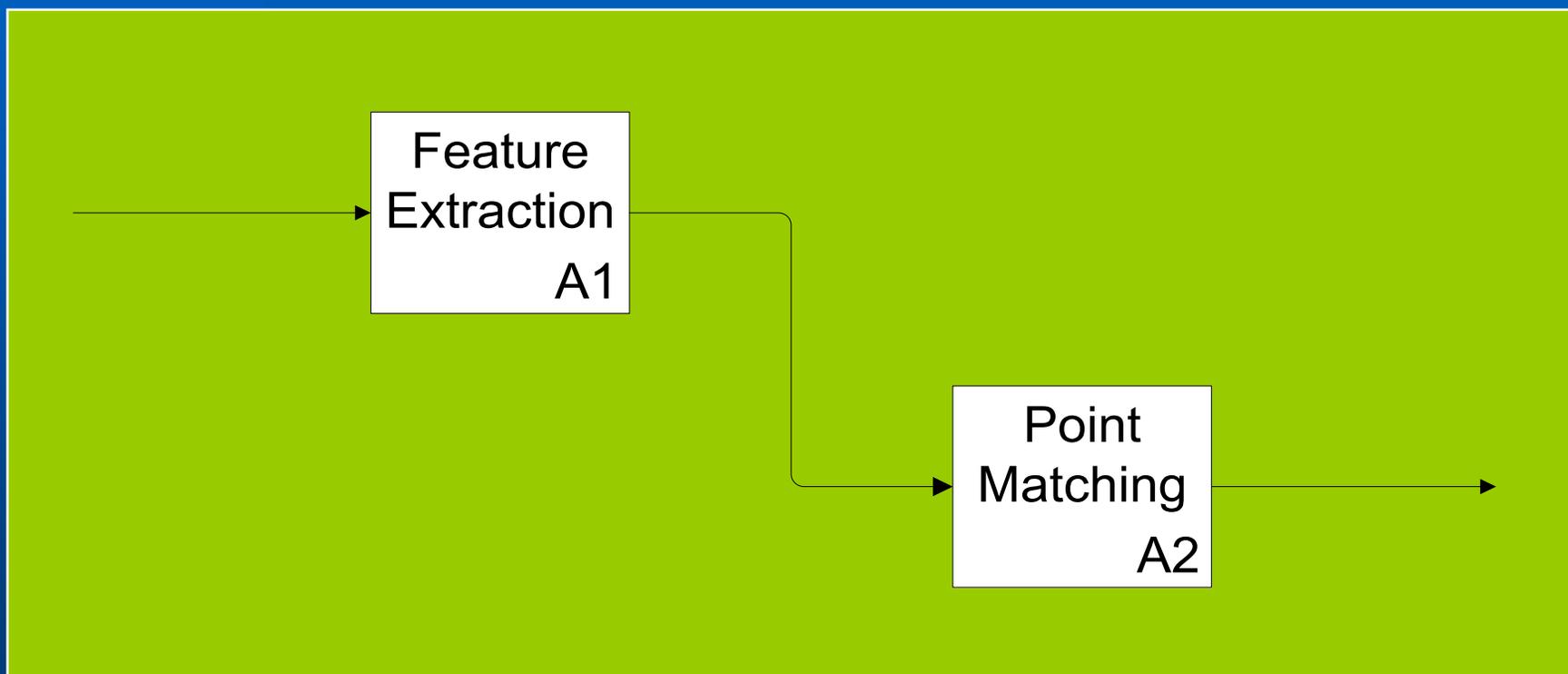


# 指紋辨識晶片系統規格定義





# 系統階層與模組設計

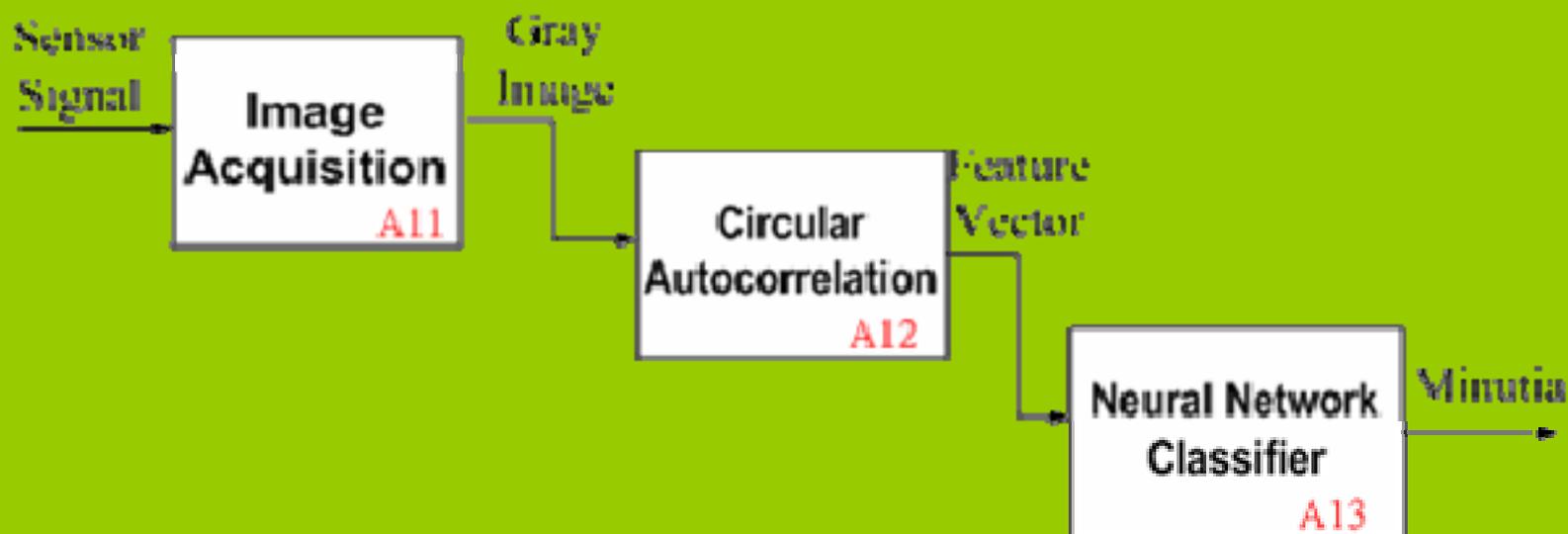


Gray Image

Mi

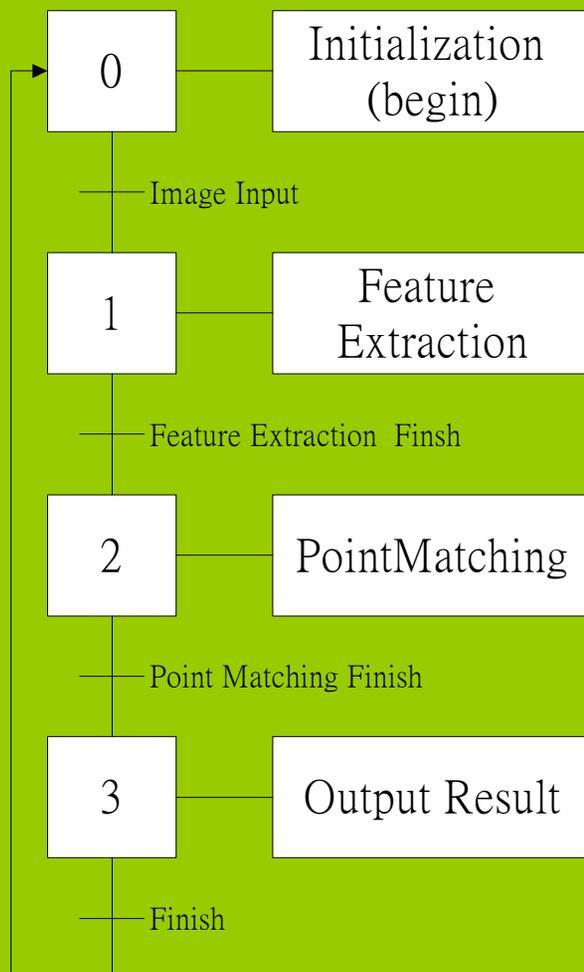


# 系統階層與模組設計





# GRAFCET建模





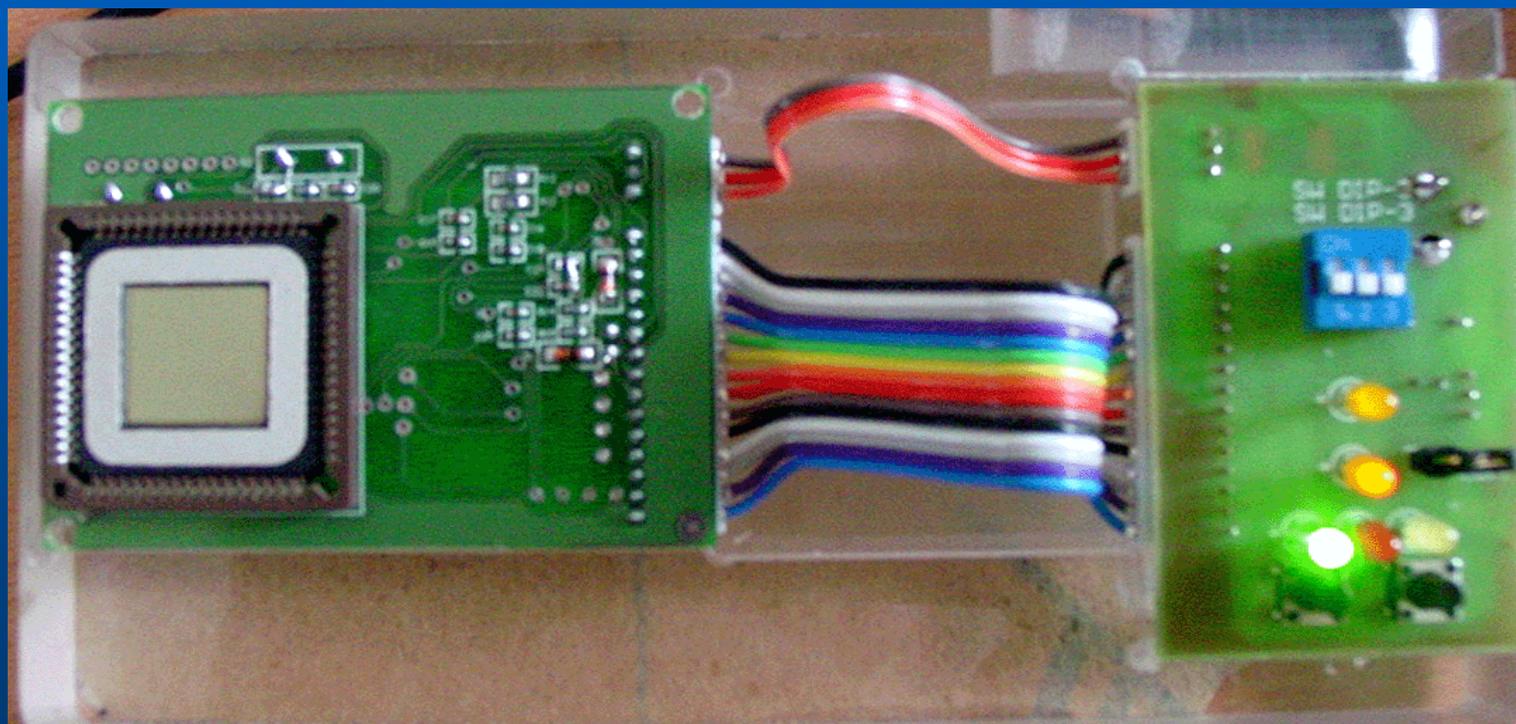


# 指紋辨識晶片界面

| AG2           |                   |
|---------------|-------------------|
| CLK           | A                 |
| RST           | NOTMATCH          |
| S             | MATCH             |
| NB1_0[5..0]   | AG23HISGM_SR      |
| NB2_0[5..0]   | AG23HISGM_Y[5..0] |
| TP2X_0[7..0]  | AG23HISGM_X[5..0] |
| TP2Y_0[7..0]  | AG22P_SR          |
| TANG2_0[7..0] | AG22P2ADDR[5..0]  |
| TYPE2_0       | AG22P1ADDR[5..0]  |
| P2X_0[7..0]   | AG22HISGM_SR      |
| P2Y_0[7..0]   | AG22HISGM_SW      |
| ANG2_0[7..0]  | AG22HISGM_I[2..0] |
| TP2X_A        | AG22HISGM_Y[5..0] |
| TP2Y_A        | AG22HISGM_X[5..0] |
| TANG2_A       | AG21P2X_SW        |
| P2X_A         | AG21P2Y_SW        |
| P2Y_A         | AG21ANG2_SW       |
| ANG2_A        | AG21P2X_0[7..0]   |
| HISGM_0[2..0] | AG21P2Y_0[7..0]   |
| HISGM_A       | AG21ANG2_0[7..0]  |
| P1X_0[7..0]   | AG21P2ADDR[5..0]  |
| P1Y_0[7..0]   | AG21TP2X_SR       |
| ANG1_0[7..0]  | AG21TP2Y_SR       |
| TYPE1_0       | AG21TANG2_SR      |
| P_A           |                   |



# 指紋辨識晶片應用模組展示





# MIAT方法論的其他應用

「指紋辨識系統晶片」

「渾沌加解密晶片」

「模糊系統(控制器與分類器)晶片」

「神經網路分類器晶片」

「神經網路影像插補器」

「8051 微控器晶片」

「SV-PWM晶片」

「FIR濾波器晶片」

「型態學平行影像處理器晶片」

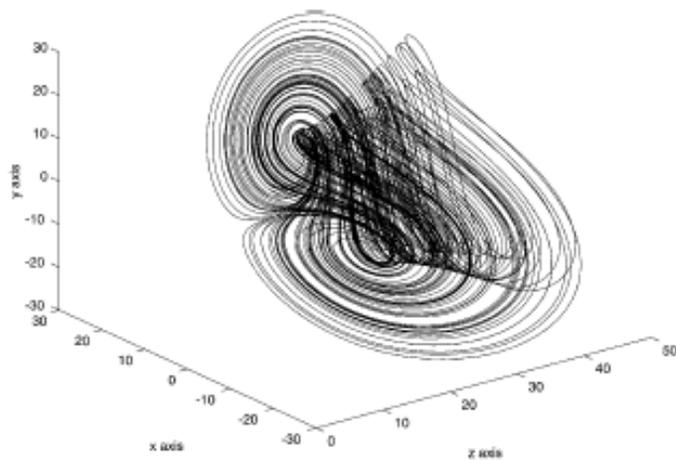
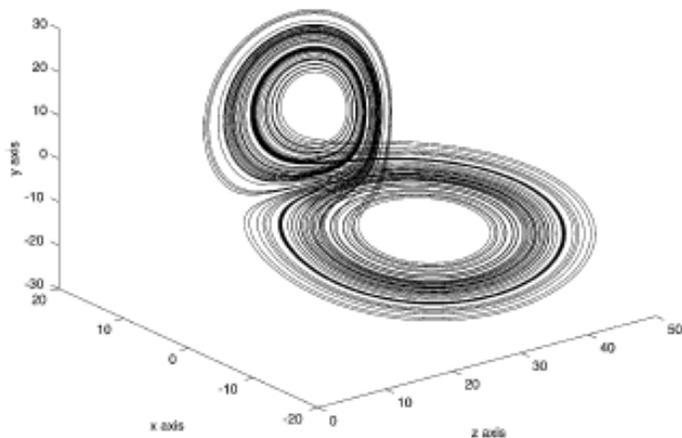


## 渾沌加解密晶片

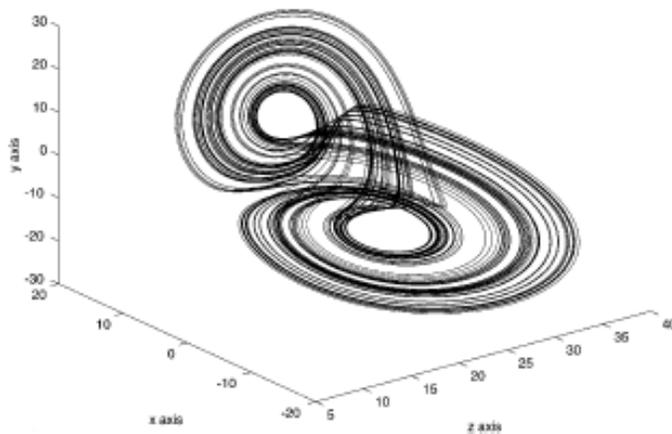
採用無法被破解的韋南(Vernam)加解密技術的概念，藉由一個仿生物行為的渾沌系統，來產生與明文資料相同長度的絕對亂數的密碼鑰匙，每一次使用的密碼鑰匙都由系統自動產生。由於渾沌系統對於初始條件的高度敏感，演化初值的微小變異會導致完全不同的演化方向，進而確保了密碼的不可預測性。



# 渾沌生成動態密碼

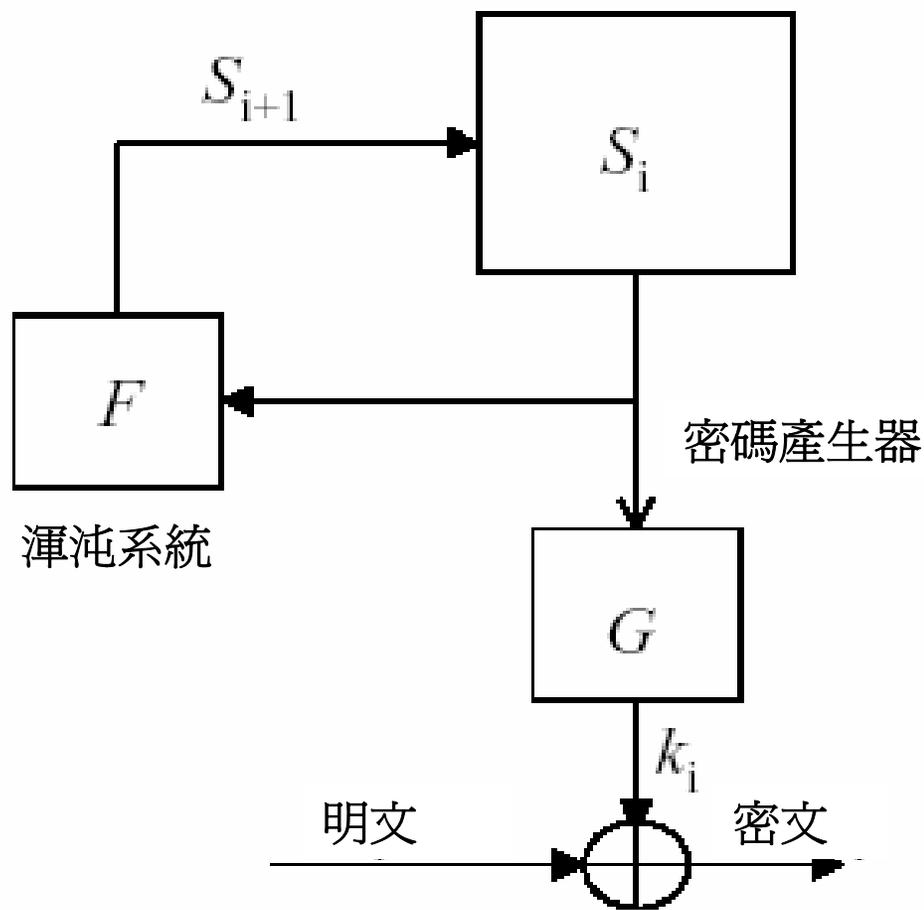


$$\begin{cases} \dot{x} = a(y - x), \\ \dot{y} = (c - a)x - xz + cy, \\ \dot{z} = xy - bz, \end{cases}$$



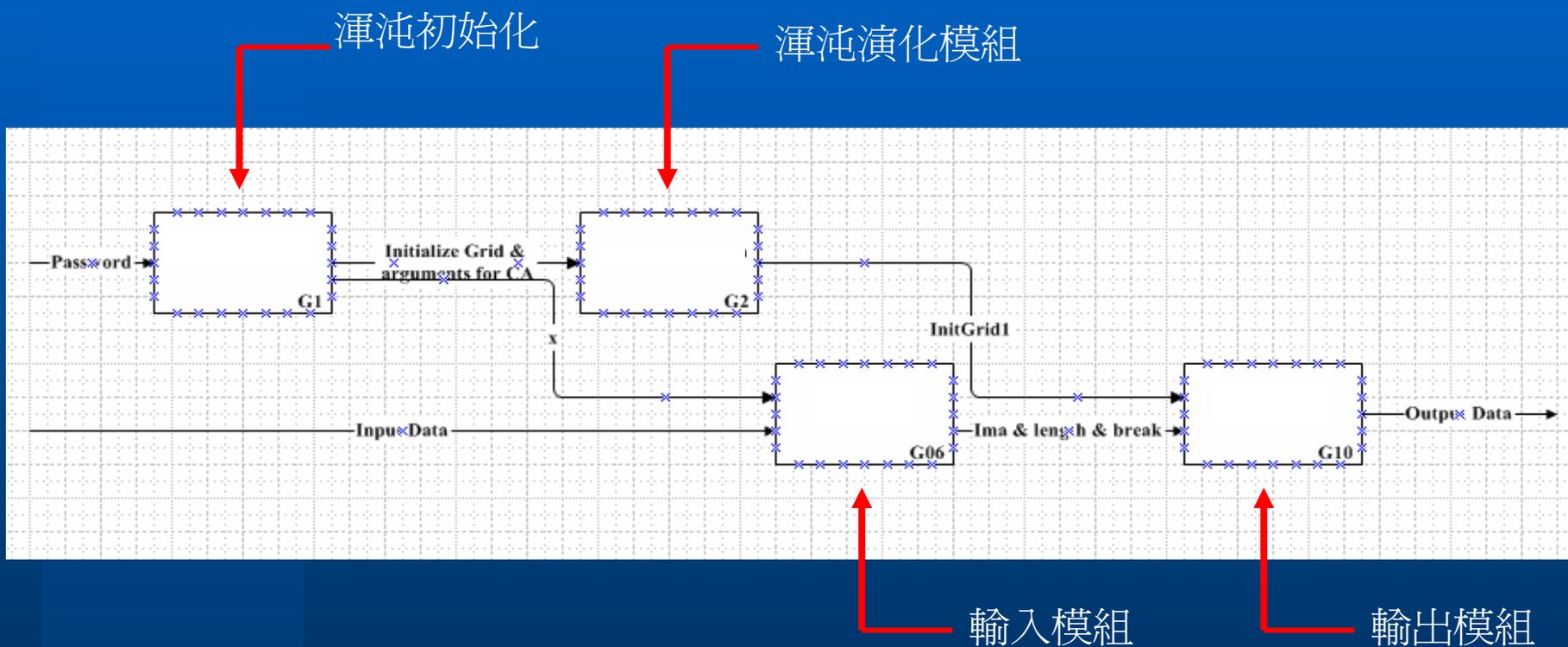


# 渾沌加密原理





# 加解密晶片系統架構





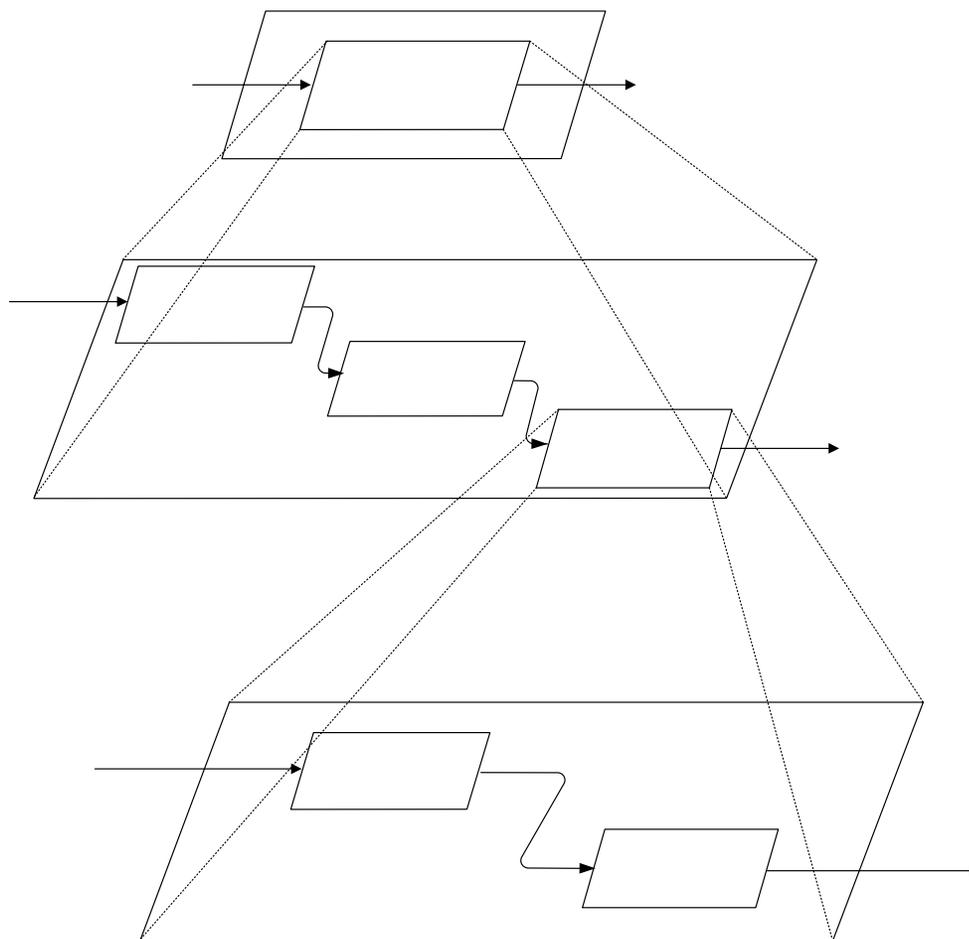
# 合成電路方塊圖

G1子模組控制器





# 模糊控制器晶片

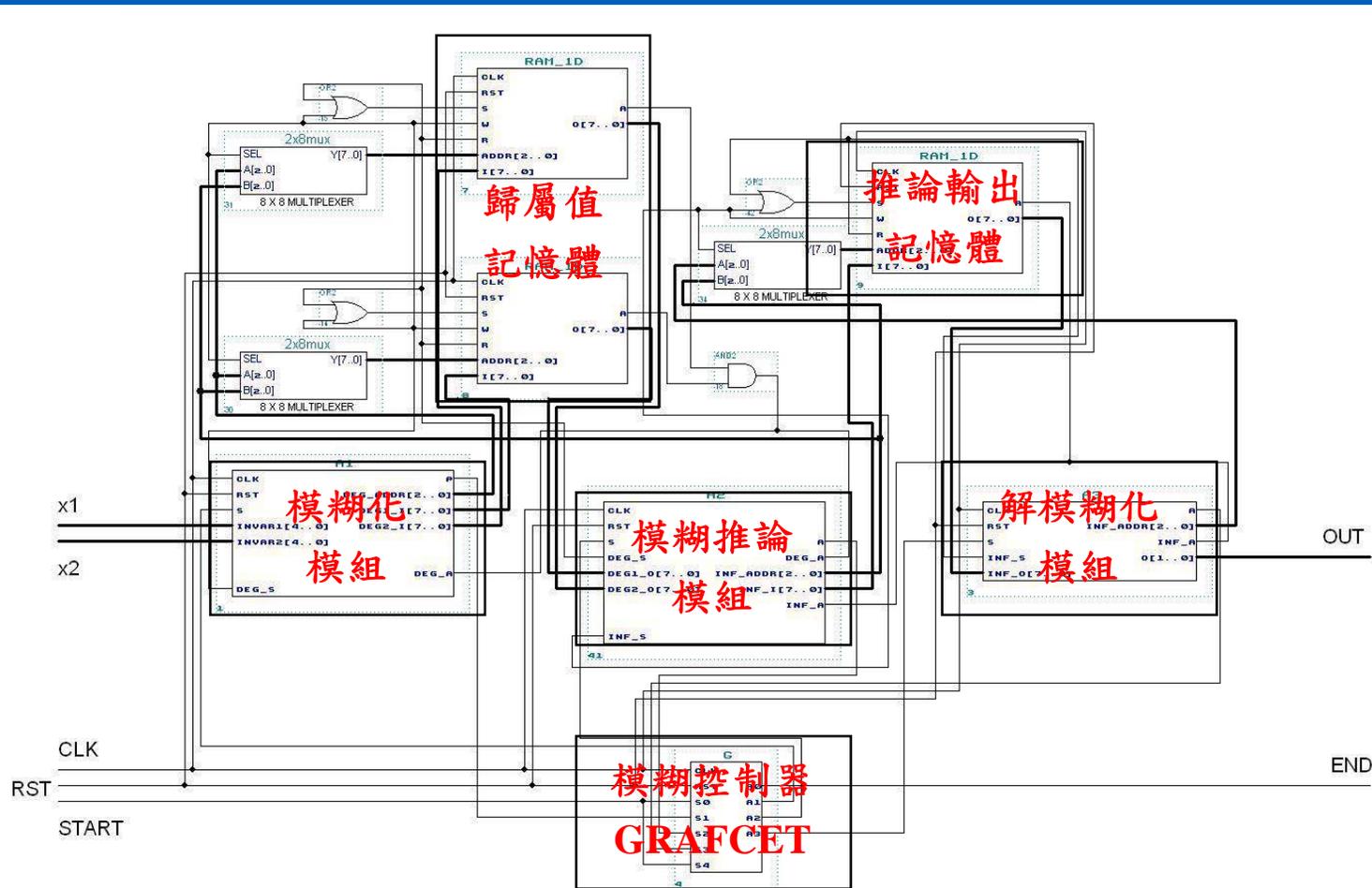


輸入

模糊



# 模糊控制器電路合成





# 模糊控制器晶片性能評估

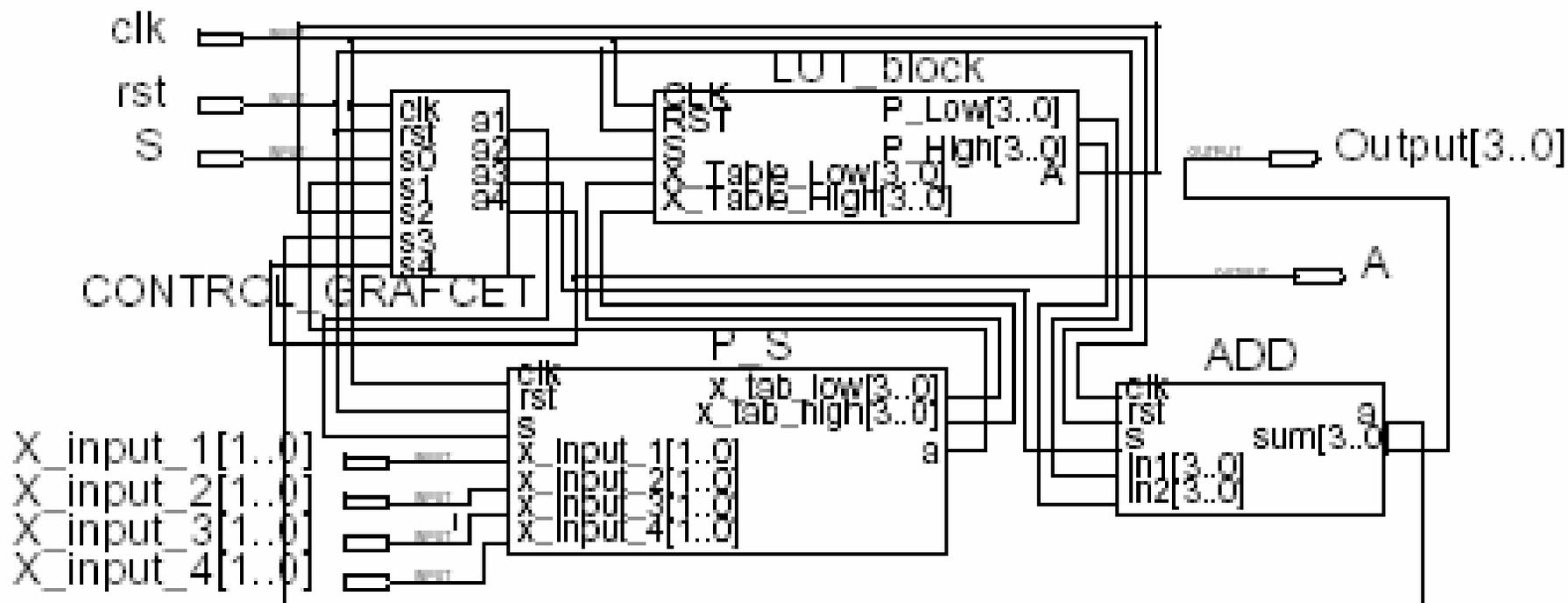
使用Altera Flex 10K70RC240的FPGA(具有70000個邏輯閘，分成3744個Logic Cell)

| 輸入值精確度 | 運算值精確度 | 使用記憶容量   | 使用邏輯單元 | 最大穩態誤差 |
|--------|--------|----------|--------|--------|
| 6 bit  | 8 bit  | 9344 bit | 32.40% | 0.007  |
| 6 bit  | 7 bit  | 8176 bit | 27.60% | 0.008  |
| 6 bit  | 6 bit  | 7008 bit | 24.20% | 0.008  |
| 6 bit  | 5 bit  | 5840 bit | 20.20% | 0.031  |
| 5 bit  | 8 bit  | 5248 bit | 32.30% | 0.015  |
| 5 bit  | 7 bit  | 4504 bit | 27.40% | 0.016  |
| 5 bit  | 6 bit  | 3936 bit | 24.10% | 0.016  |





# FIR 電路合成





## 結語

- 本方法論是基於Top-Down設計範式(paradigm)，系統設計採階層式、模組化的功能架構。
  - 針對每一個功能獨立的模組，使用GRAFCET圖形化工具建立其離散事件模型。
  - 根據一組合成的法則，所有GRAFCET模型得以轉譯為VHDL控制器電路，再結合一些通用的組合邏輯元件，便可實現系統的高階合成。
- 此一方法論將不同的方法、技術、工具和模型，整合成一個嚴謹的、完整的設計與自動合成程序，有利於研發團隊從事群體的、分工式的複雜系統設計工作。



# 未來發展

- 元件資源配置最佳化和使用排程
- 分散式控制與非同步電路設計
- Grafcet-based 智慧型系統核心IP開發
- 高階合成 CAD工具建置