

# 消費性電子產品研發管理 和設計方法論

Product R & D Management and Design  
Methodology in Consumer Electronic

陳慶瀚

機器智慧與自動化技術(MIAT)實驗室

義守大學電機系

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

2005年3月11日



# 大綱

PART 1 :消費性電子市場與產品設計

PART 2 :消費性電子產品研發管理

PART 3 : MIAT方法論



# PART 1

## 消費性電子市場與產品設計



# 消費性電子產品有哪些？





# 消費性電子產品的特性

- 技術推陳出新
- 大量生產形式
- 產品生命週期短
- 產品單價低
- 內建一個以上的微處理器



# 消費性電子產品類別

- Storage Media
- Car Stereo & Navigation
- Hand-Held
- Home & Portable Audio
- Telephone & Home Office
- Video
- Electronic Gaming
- Accessories

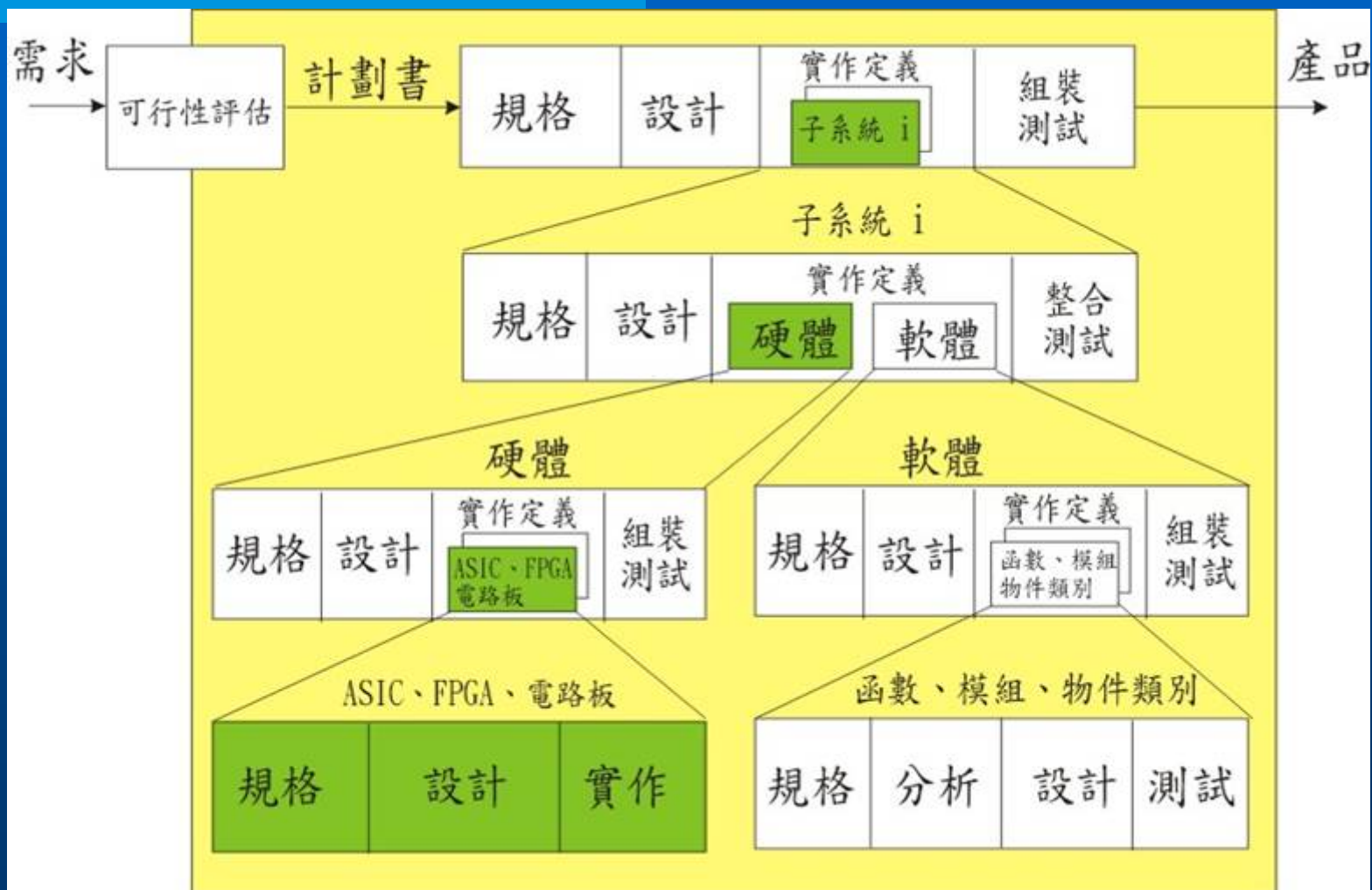


## 2005年主要消費性電子產品

- Flat TVs
- DVD players
- Home theater systems
- In-car entertainment
- Digital still cameras
- MP3 players
- Personal Digital Assistants
- Household appliances



# 消費性電子產品設計流程





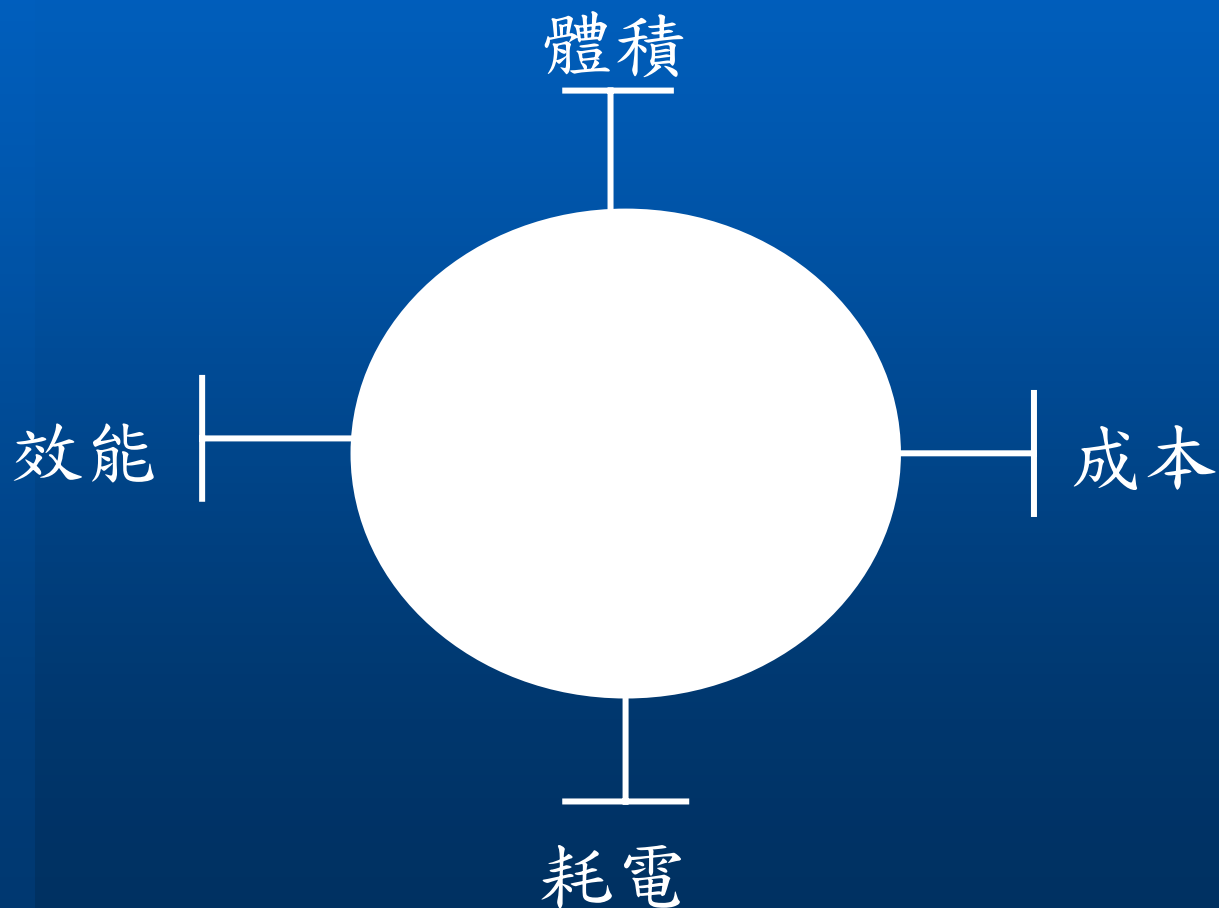


# 消費性電子產品設計挑戰

- 功能正確性
- 單位製造成本
- NRE(Non-Recurring Engineering)費用
- 體積
- 效能
- 耗電
- 彈性
- 原型產品產出時間(Time-to-prototype)
- 上市時間(Time-to-market)
- 維修性
- 安全性

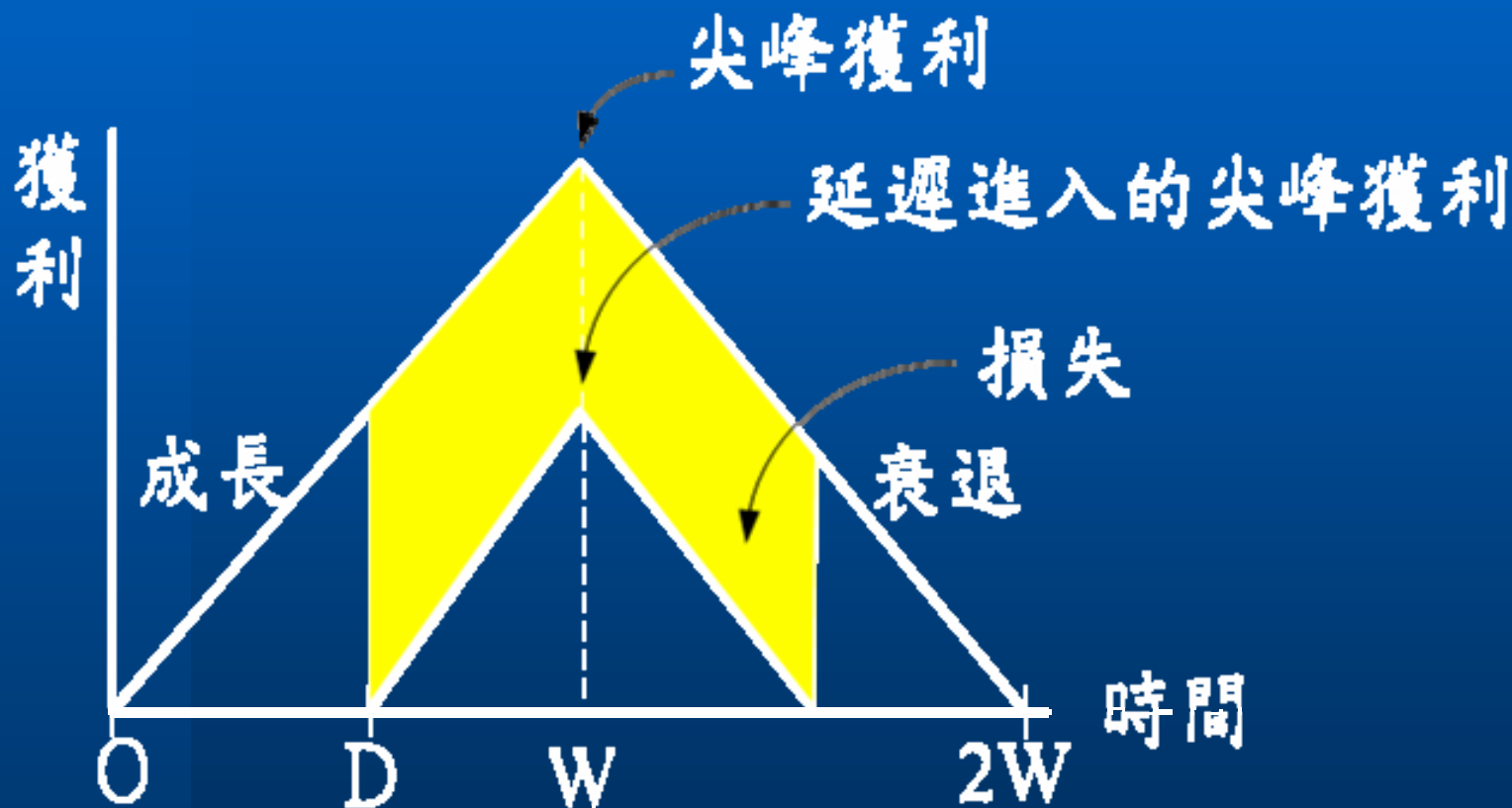


# 消費性電子產品設計挑戰





# Time-to-market





# Delay Entry的損失計算

- 產品生命週期 $2W=52$ 周, 延遲進入4周 $D=4$
- $(4*(3*26 - 4)/2*26^2) = 22\%$
- 產品生命週期 $2W=52$ 周, 延遲進入 $D=10$
- $(10*(3*26 - 10)/2*26^2) = 50\%$



# NRE成本和單位成本

- 單位成本: 每一個產品樣本的平均製造成本
- NRE 成本: 產品研發的一次性成本
- 總成本 = NRE成本 + 單位成本 \* 產品數量
- 最終產品成本 = 總成本 / 產品數量

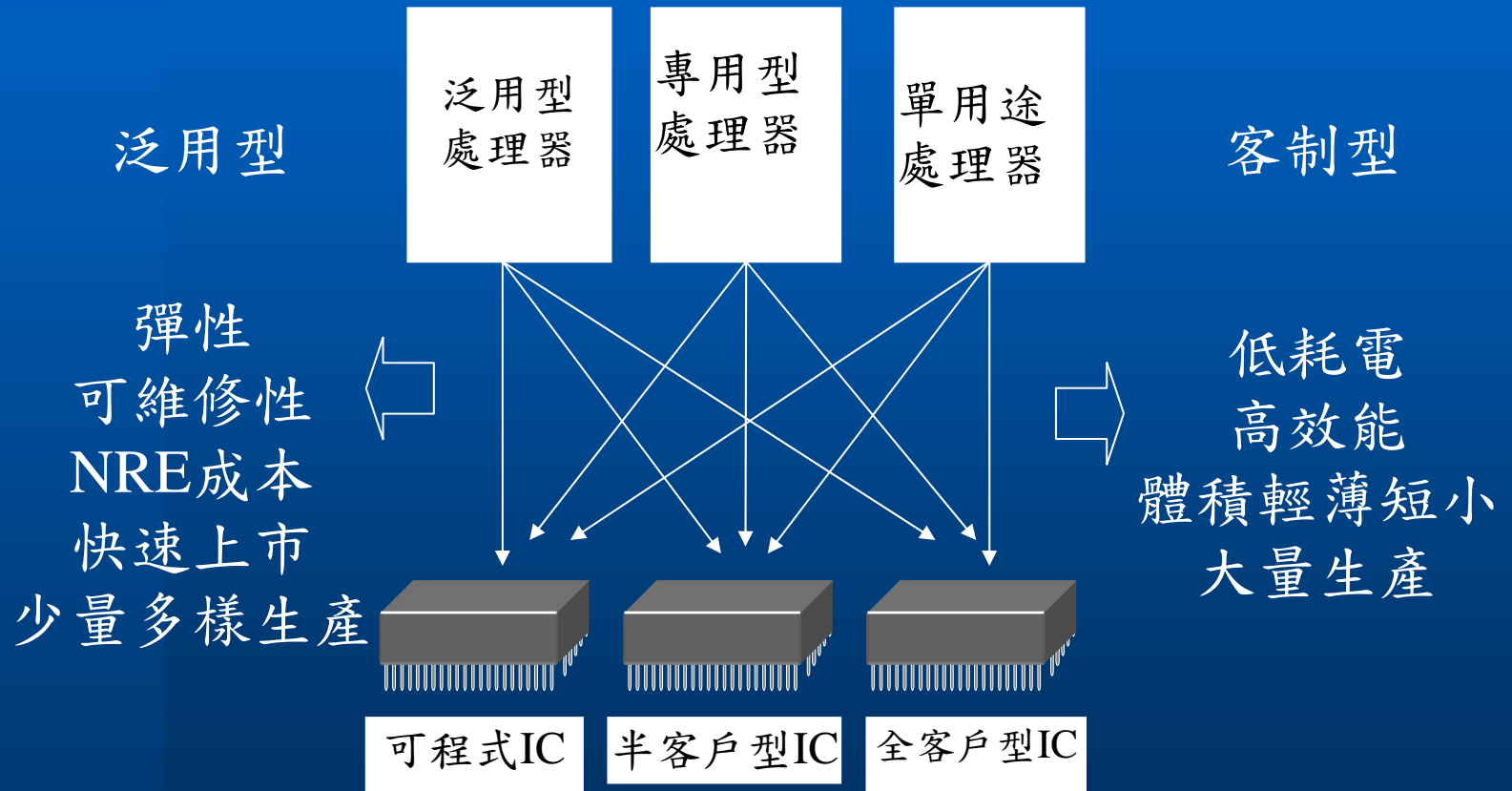


# 消費性電子產品主要技術

- 微處理器(Microprocessor): 泛用型處理器, 專用型處理器
- 晶片(IC): 全客戶型IC, 半客戶型IC, 可程式IC
- 設計方法論(Design Methodology): 設計、編譯、電路合成、測試、驗證的工具、技術和方法



# 設計 Trade-off





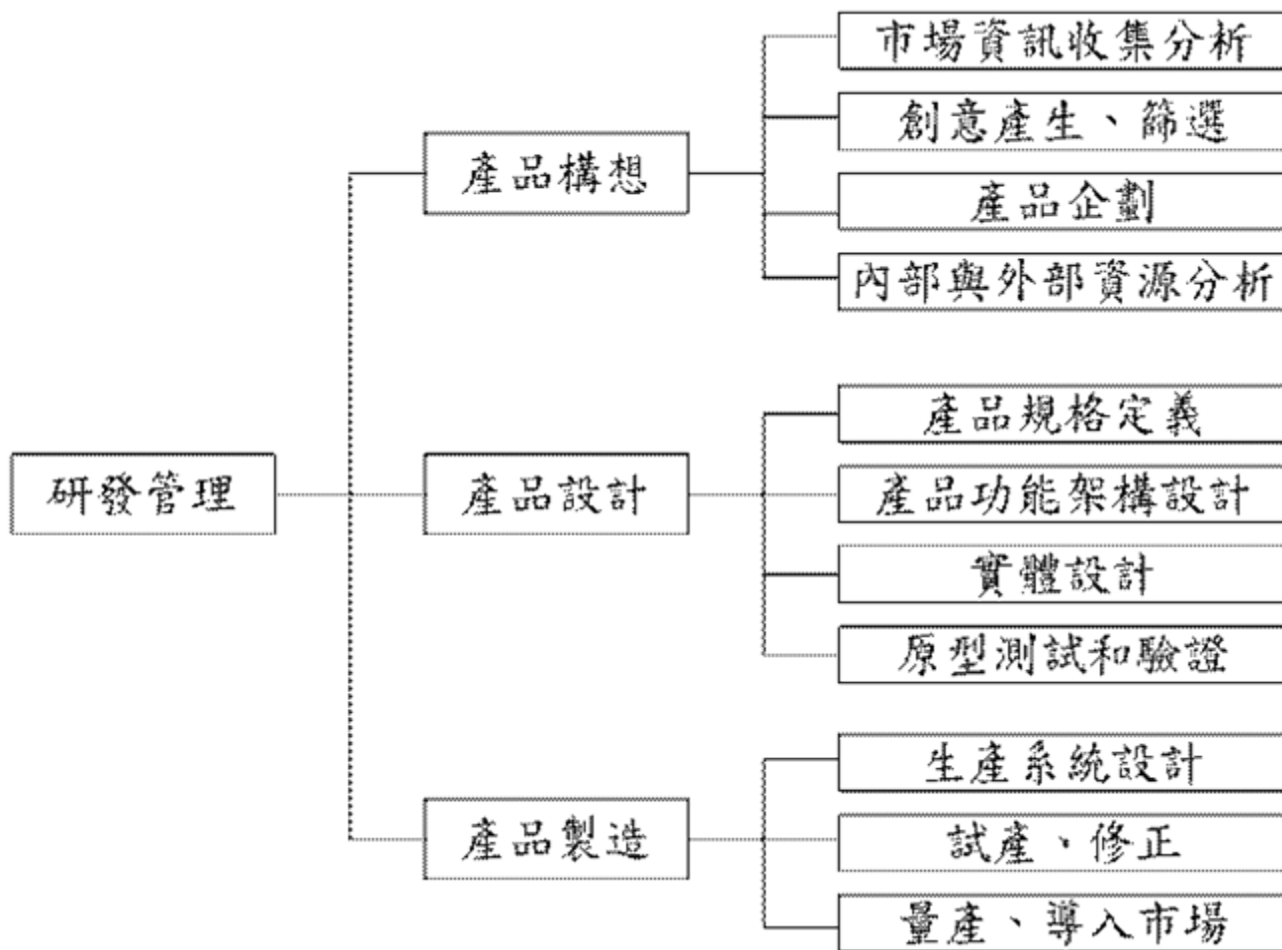
## PART 2

# 消費性電子產品研發管理





# 研發管理的任務和工作流程



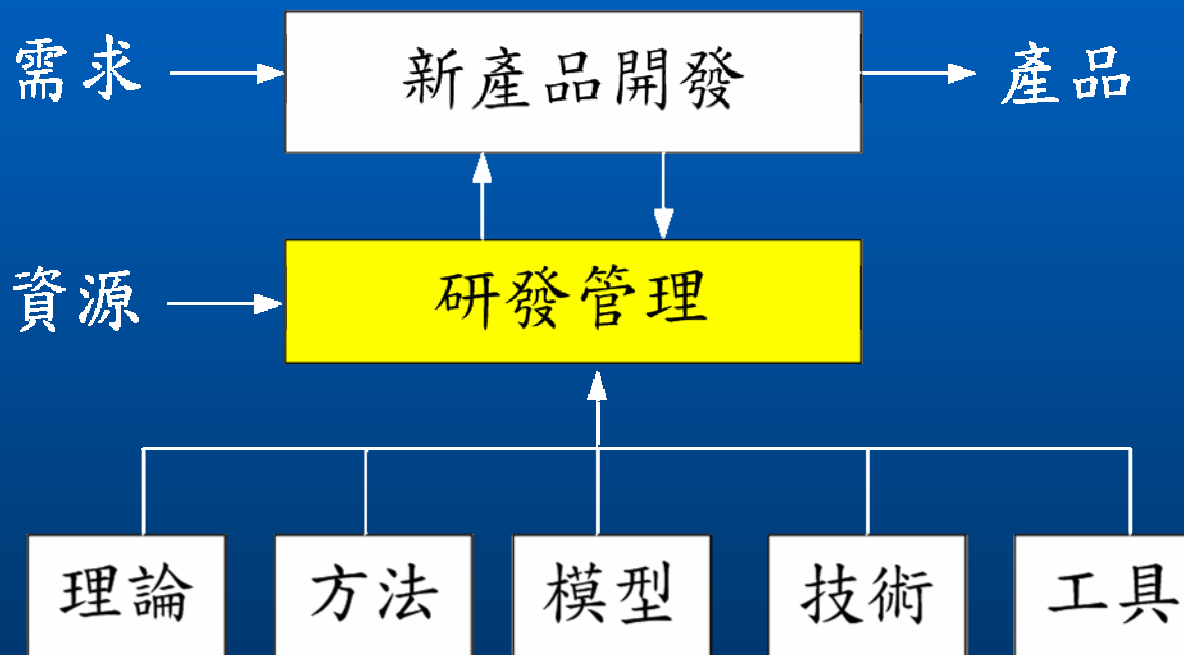


# 研發管理的挑戰

任務	挑戰
專案管理	研發控管不易
文件管理	檔案分類、搜尋不易；版本混亂
知識管理	缺乏知識分享和利用平台
溝通	溝通時效和溝通成果偏低
協同作業	工作模塊的切割、聯繫和整合困難

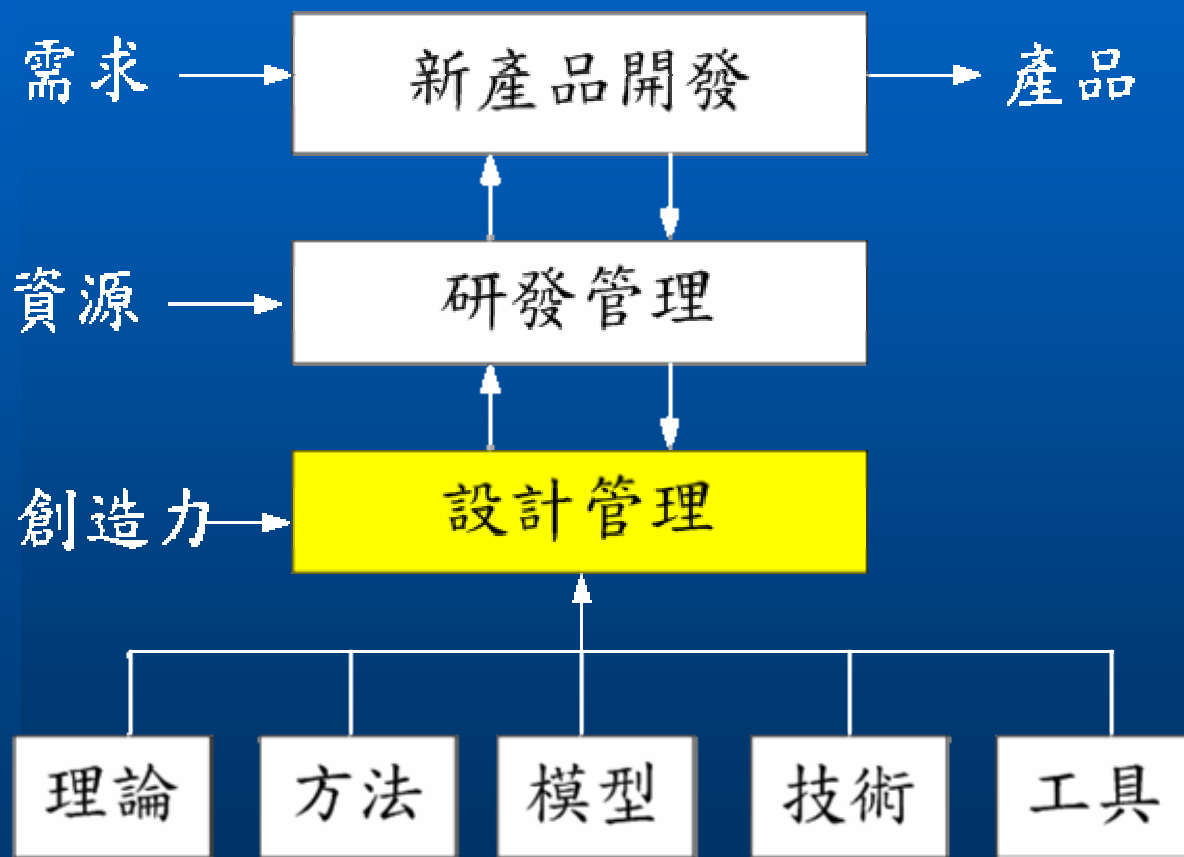


# 新產品開發鏈與研發管理



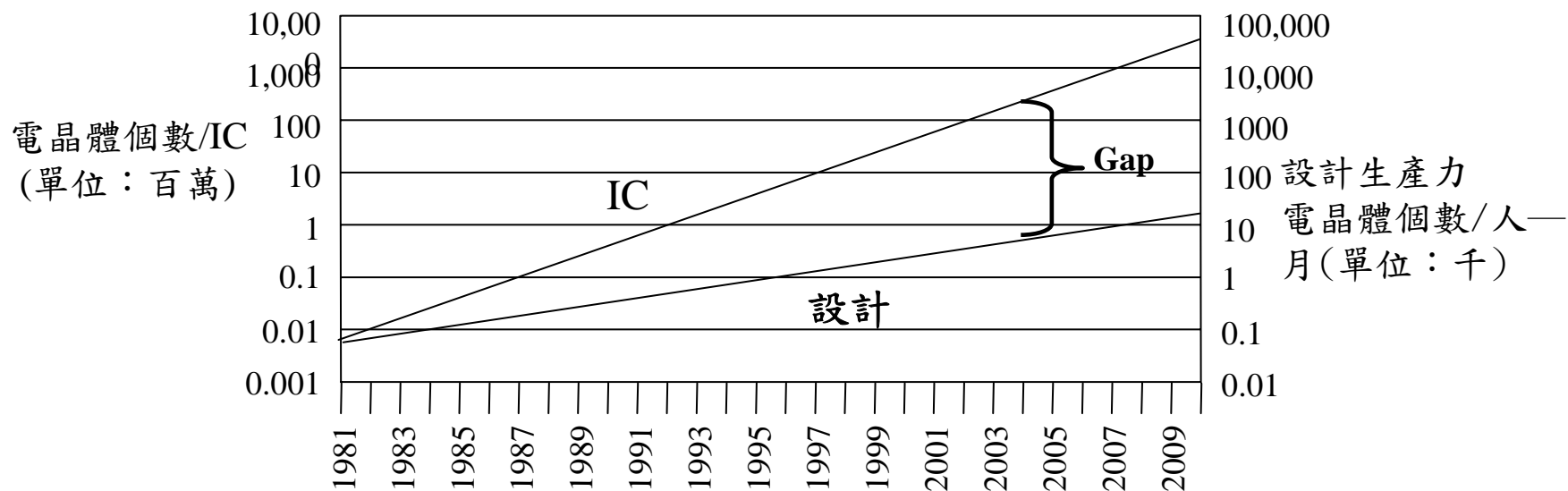


# 開發鏈的新角色—設計管理



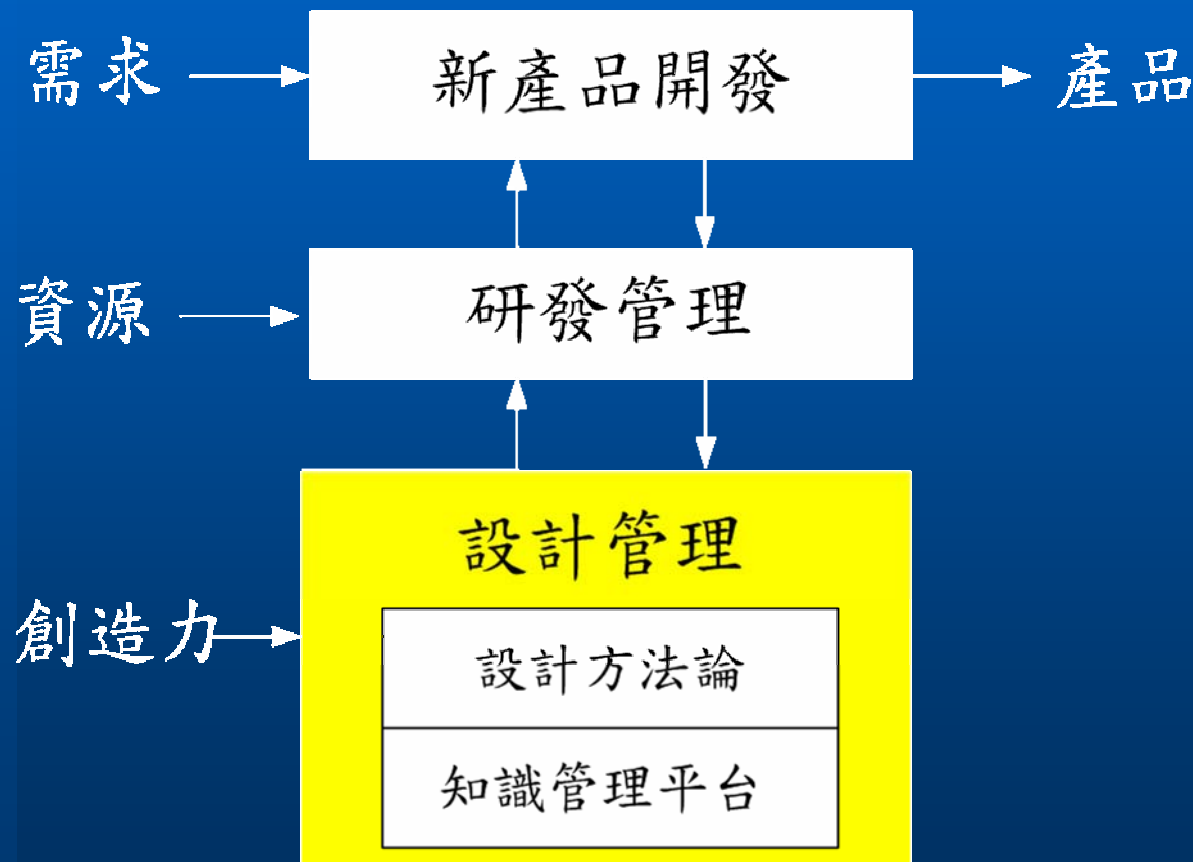


# 為何需要設計管理？





# 如何進行設計管理？





# 設計知識的環境

## 設計知識環境

參考文件書籍

廠商技術支援

程式碼

論文

第三方解決方案

設計案例

其他人員經驗

訓練課程

競爭者網站

網路資源



# 建構知識管理平台的重要性

- 使個體的經驗和知識得以累積和分享；
- 研發的知識就是財產，因此知識管理平台在於避免公司資產遭到棄置；
- 運用群體知識，避免個人知識解題的侷限性；
- 避免重蹈錯誤和資源重複投資。





# 知識管理的實務與困難

## 知識管理的實務：

- 檔案伺服器文件管理；
- 關鍵字搜尋引擎；
- 公司內部入口網站；
- 制定標準、綱領手冊

## 知識管理的困難：

- 如何在適當時機將適當檔案交付研發人員？
- 知識的異質性和不連貫性導致溝通、分享的困難；
- 研發人員/團隊不想在手冊尋找答案；
- 分散的知識內容無法整合、累積、和演繹新的知識。

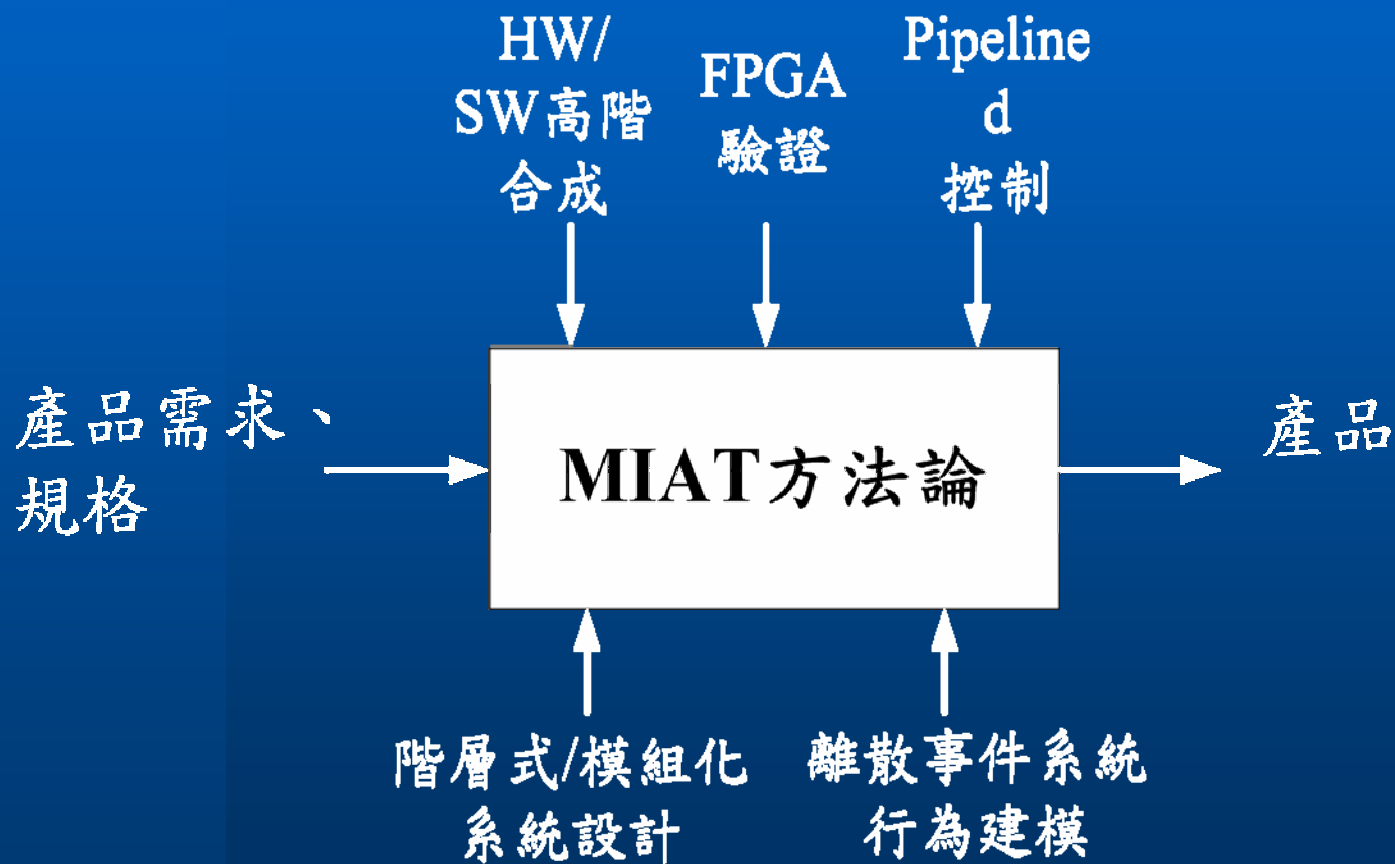


# PART 3

## MIAT設計方法論



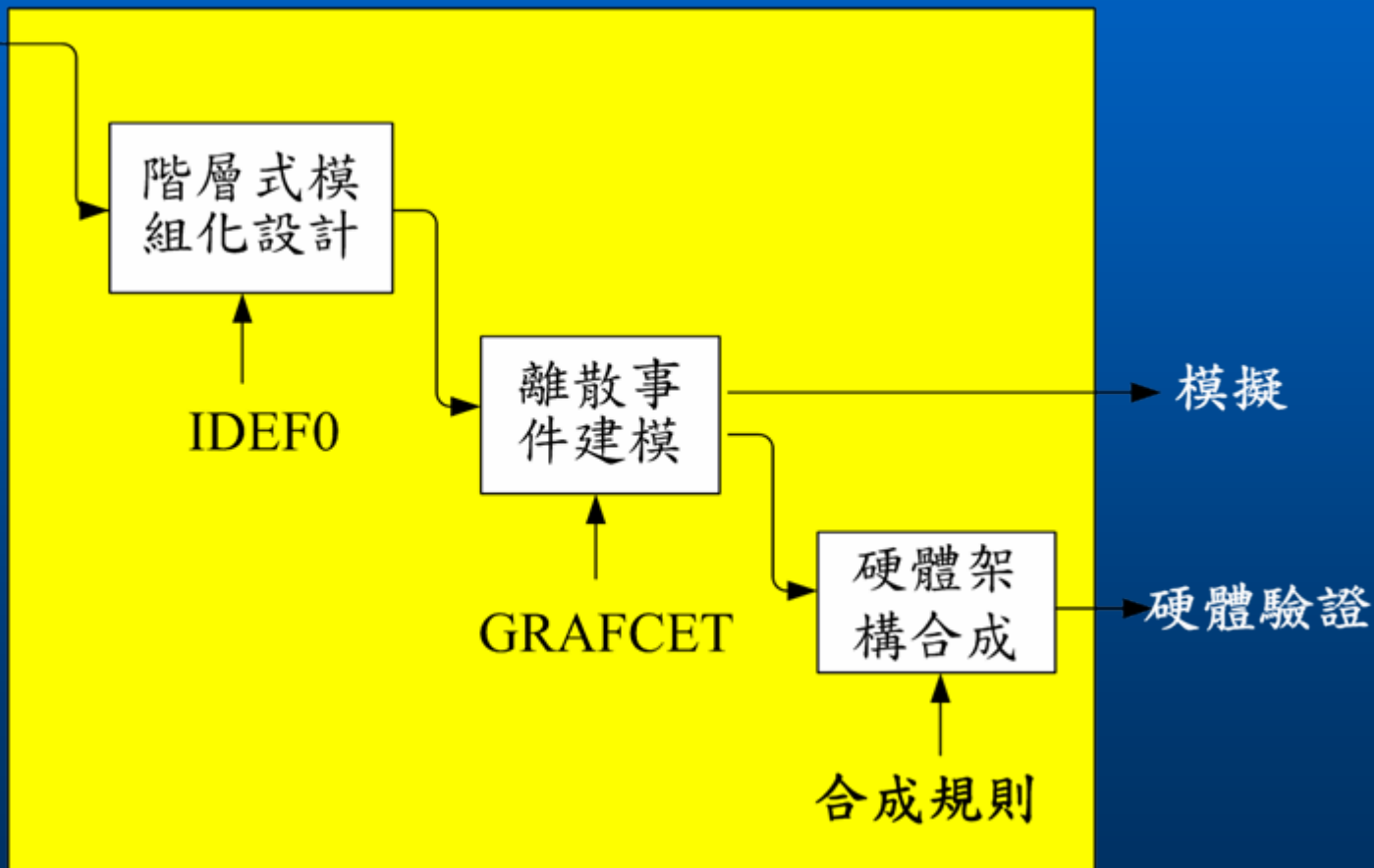
# MIAT設計方法論





# MIAT設計方法論架構

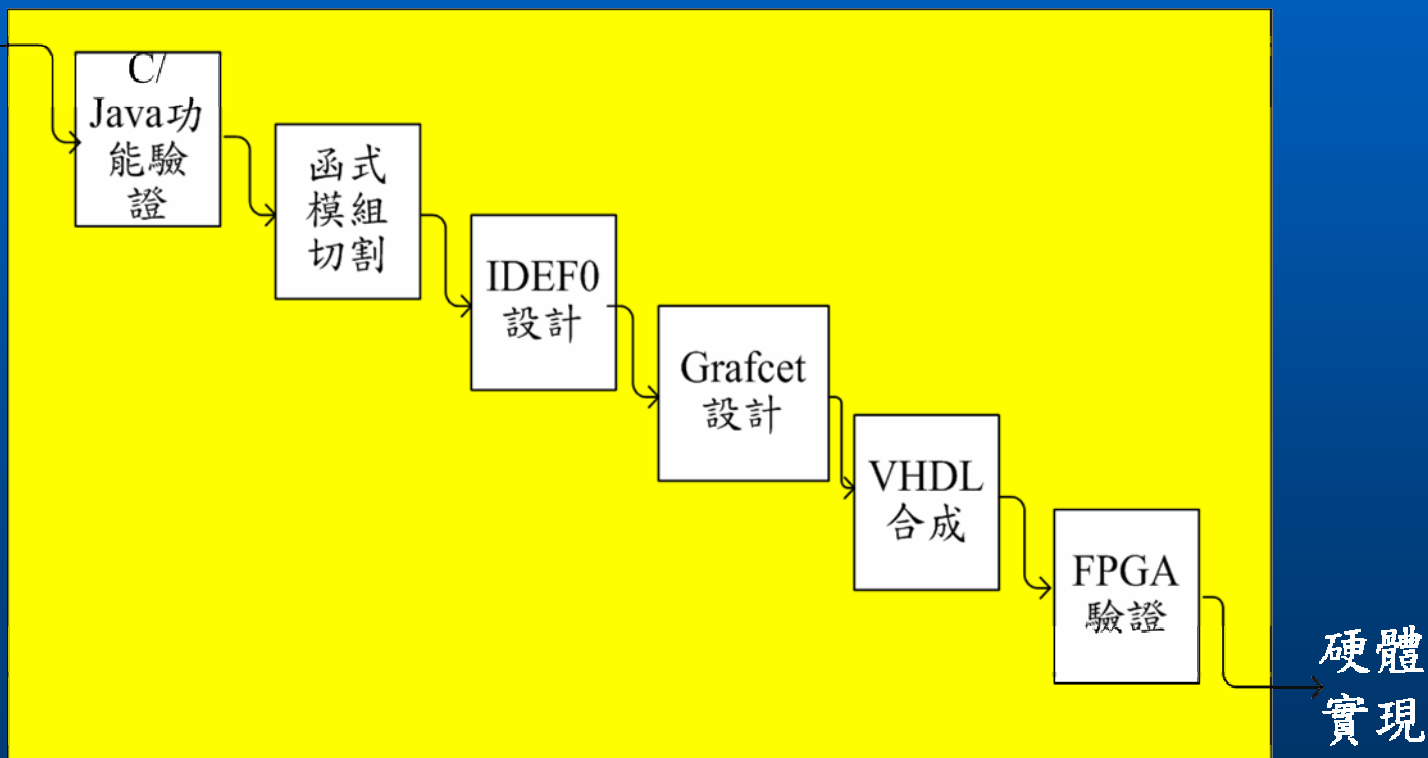
產品規格  
定義





# MIAT設計方法論的實務

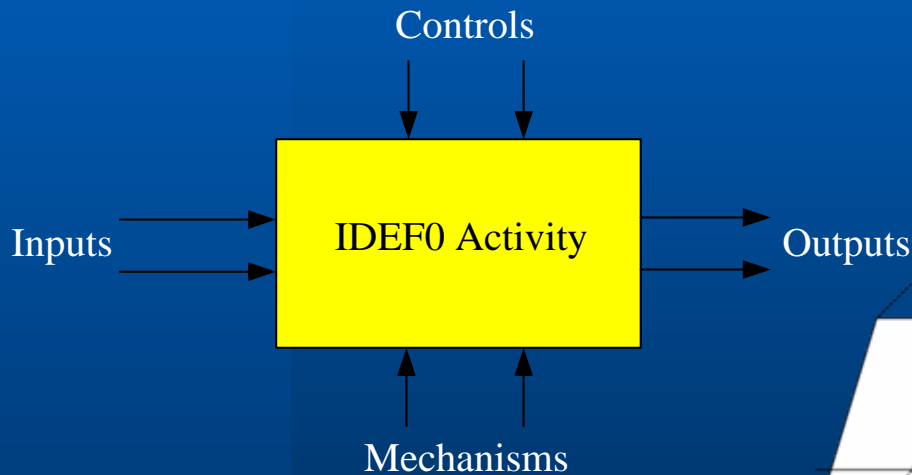
方法、  
演算法



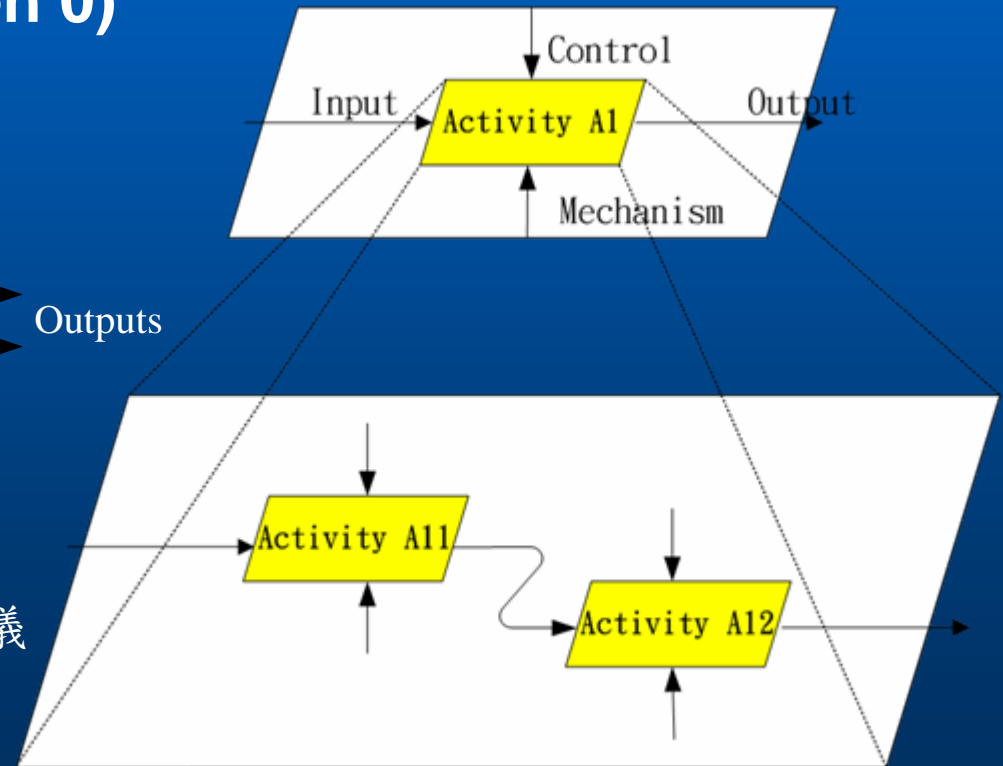


# 階層式與模組化設計

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



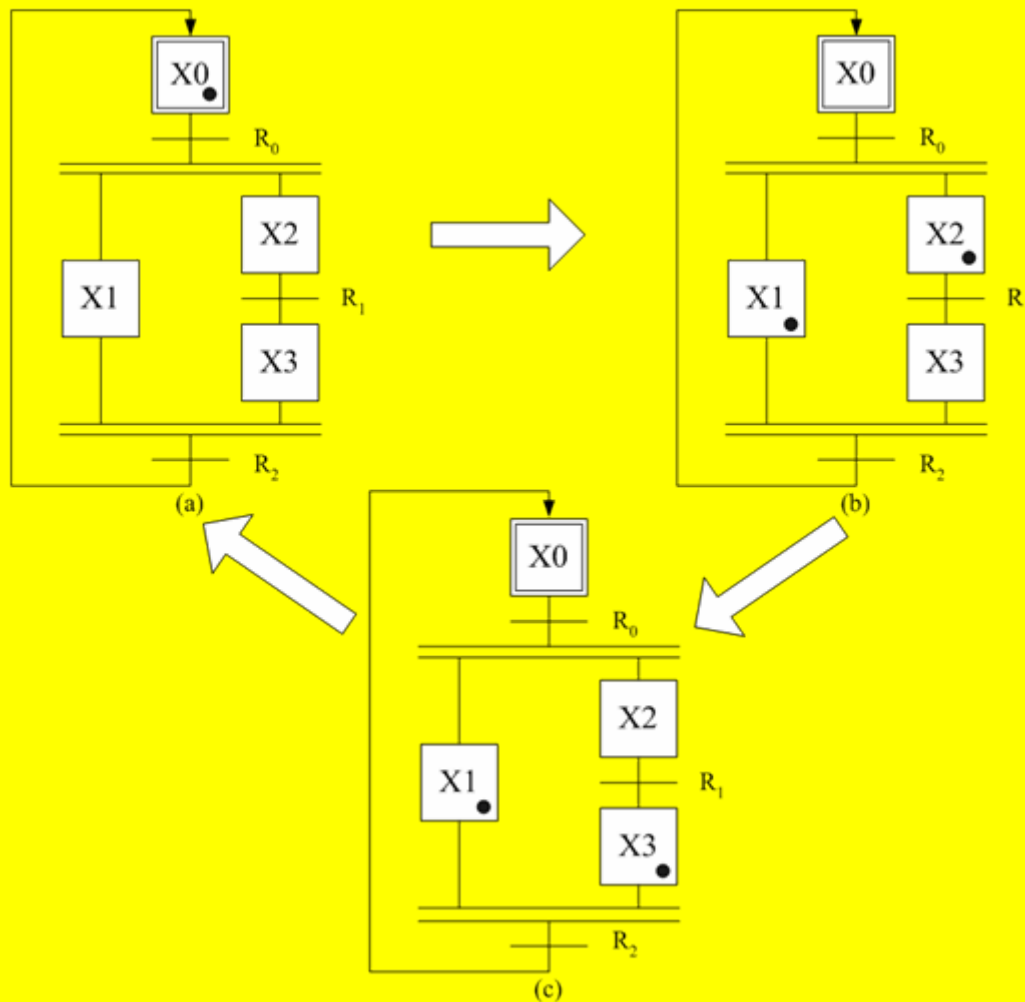
IDEF0基本模組功能方塊與箭頭意義



IDEF0 階層化架構



# Grafcet的離散事件模擬





# VHDL硬體合成

```
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';

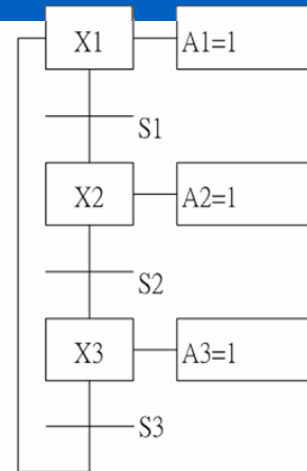
    elsif CLK'EVENT and CLK='1' then

      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;

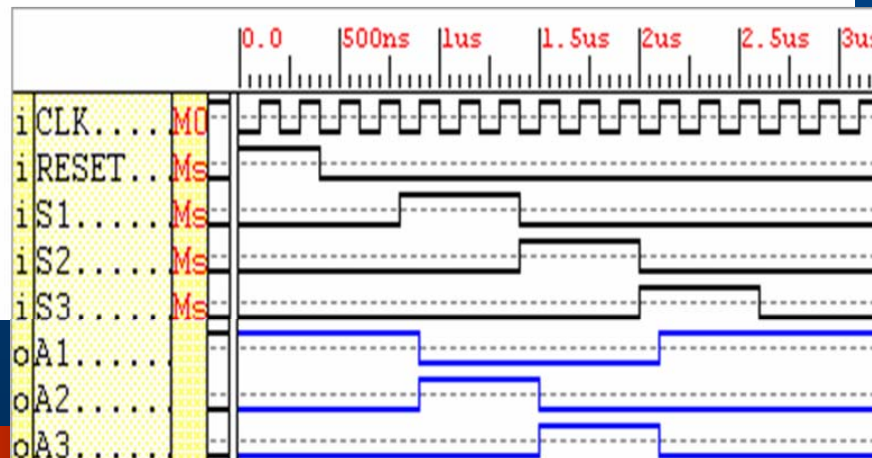
    end if;

    A1<=X1;
    A2<=X2;
    A3<=X3;

  end process;
end DEMO1_arch;
```



```
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;
```







# MIAT設計方法論特色

- MIAT設計方法論是基於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，再結合一些通用的資料流(data flow)元件，便可實現系統的高階合成；
- 方法論採用均一的解決問題程序，包括問題形式化(formalization)、建模(modeling)和設計規則、工具和技術。因此利於知識的整合和知識網路的建構。



# MIAT設計方法論在消費性電子產品 研發管理的貢獻

- 設計者得以採用系統化的方式而不需過度仰賴個人技術經驗來從事新產品設計工作
- 利於複雜系統團隊分工和協同設計
- 確保設計品質
- 縮短產品設計時間
- 高度重複使用性，易於修改和擴充原產品
- 技術和經驗得以累積，形成知識管理平台



## 結語

- 以設計管理為核心的研發管理是下一個世代消費性電子產品開發的重要議題；
- MIAT設計方法論採用階層式、模組化、top-down流程進行系統設計和高階合成，具有加速產品研發時程、確保模組的可靠度和重複使用性，並簡化系統整合的複雜度。
- MIAT設計方法論提供了一個消費性電子產品設計知識管理平台，研發團隊的所有成員得以在共同的知識底層架構之上進行知識的學習、分享、傳遞和累積。因此有助於發展以設計管理為核心的研發管理策略。