

CH3 Platform-Based SoC Design

3.1 簡介

- 再用的組件集結形成針對特定領域產品的一整組可再用之物件，我們稱之為**平台 (platform)**。
- 使用此平台所做出的差異化系統設計，我們稱之為**基於平台設計 (platform-based design)**。



圖 3.1 系統晶片中可再用的元件

- 平台中可再用的部分通常稱之為**智慧財產** (intellectual property, IP)，或稱為**智產模組**，在限定應用類型中，由設計工程師產生這些智慧財產的設計。

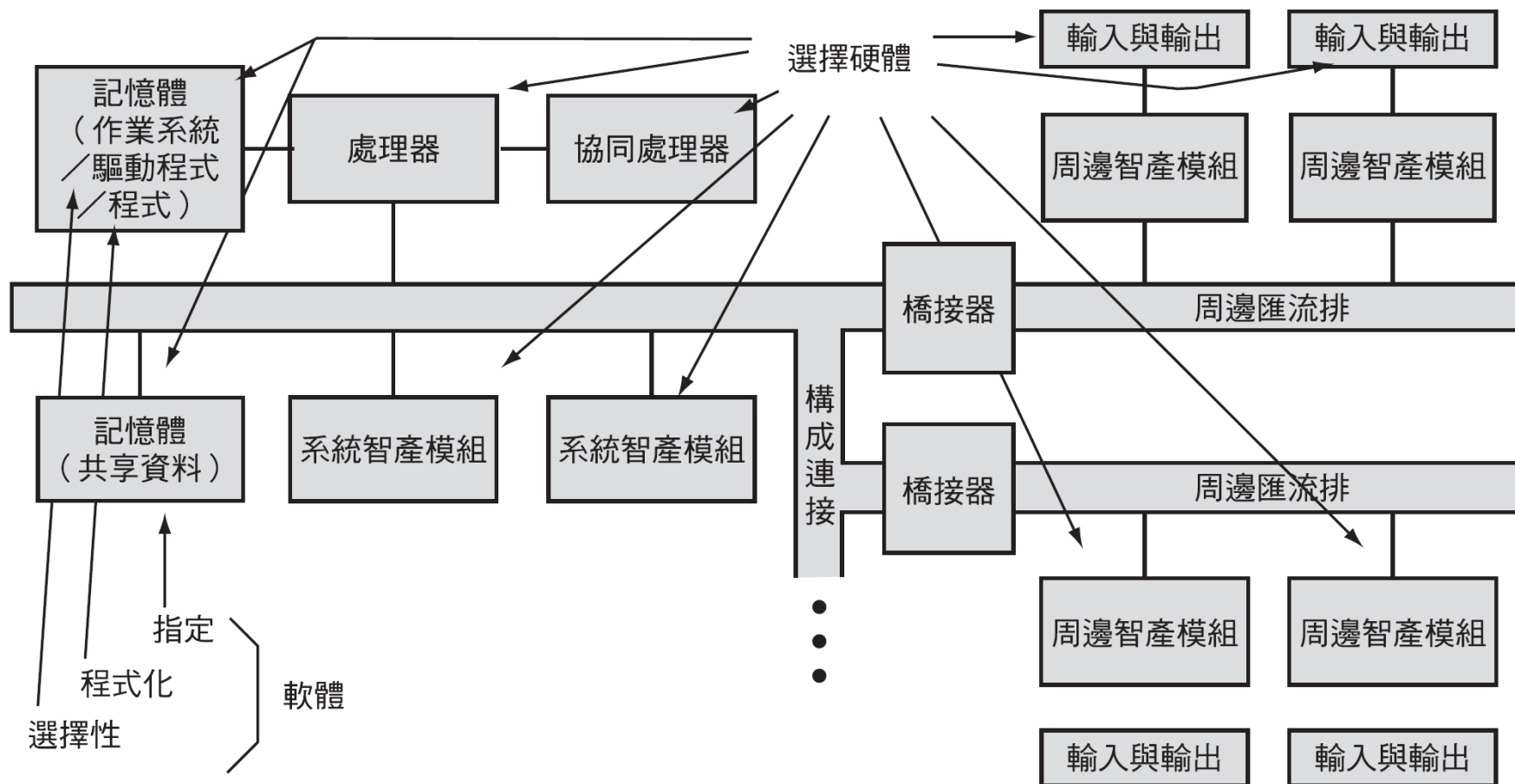


圖 3.3 基於平台設計範例

3.2 基於平台設計組成

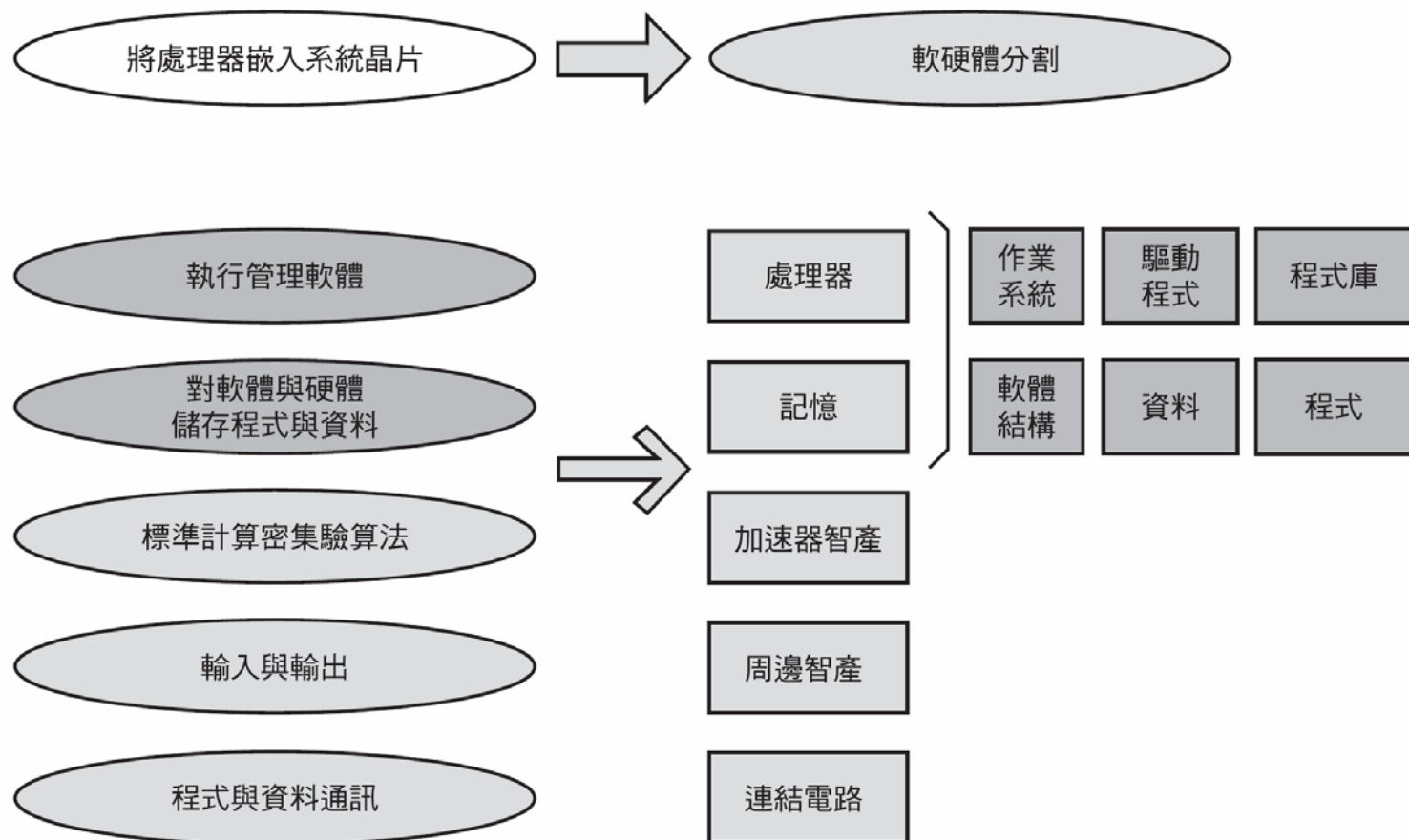


圖 3.4 將功能對映至硬體與軟體元件

- 在一組**平台**中，需要五種設計物件：
 - (1) 硬體結構以匯流排及連結電路整合處理器、記憶結構與其他硬體元件。
 - (2) 硬體智產 (**hardware IP**) 模組包括所需處理器、匯流排、記憶體、類比／數位轉換器 (**A/D converter**)、數位轉類比元件、直接記憶體存取 (**direct memory access, DMA**)、數據機 (**modem**)、編碼譯碼器 (**codec**) 等。
 - (3) 軟體結構選擇考量是否具有作業系統、虛擬記憶或實體記憶、即時排程、通訊與同步、與記憶體分配等能力。
 - (4) 軟體智產包括合適的即時作業系統 (**real-time operating system, RTOS**)、驅動程式與應用程式庫。
 - (5) 進行系統組態與評量所需的方法與支援工具。

- 在**基於平台設計**程序中，設計工程師可以進行四類活動：
 - (1) 選擇何組軟體與硬體平台元件可用於該系統設計。
 - (2) 設計特殊應用硬體元件以加速特殊應用處理。
 - (3) 系統。設計訂做之應用軟體元件以用於複雜與可的處理功能。
 - (4) 決定以何組硬體與軟體元件組態所需之應用

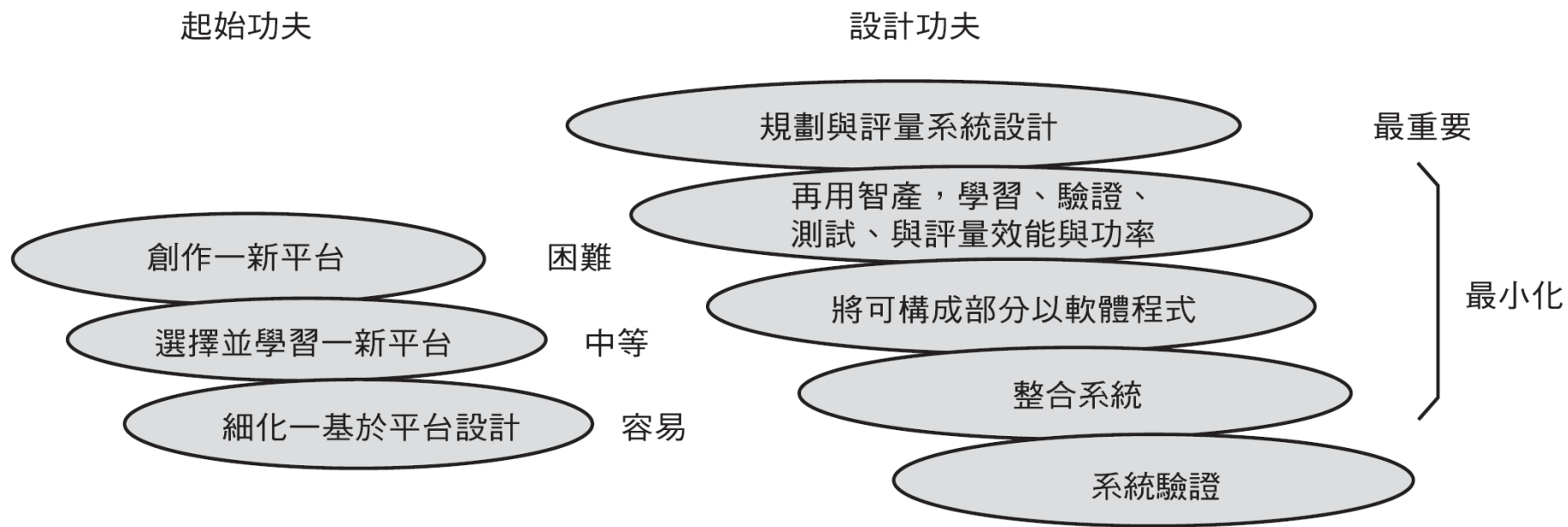


圖 3.5 各類基於平台設計工作

- 用於基於平台設計的所有階段，我們需要三種**基於平台設計驗證**工作：
 1. 硬體智產驗證
 2. 軟體驗證
 3. 系統驗證

- 系統晶片的可測試性設計 (design-for-testability, DFT) 問題可分為兩個可測試性子問題：
 1. 硬體智產模組的可測試性
 2. 整合系統設計的可測試性

3.3 基於平台設計的方法

- 基於平台設計程序由三個階段組成：
 - (1) 平台設計
 - (2) 基於平台設計
 - (3) 設計驗證

- 階段 1 之平台設計程序由四個步驟組成：
 - － 標的應用 (target application) 的分析
 - － 平台設計
 - － 平台評量
 - － 平台調整

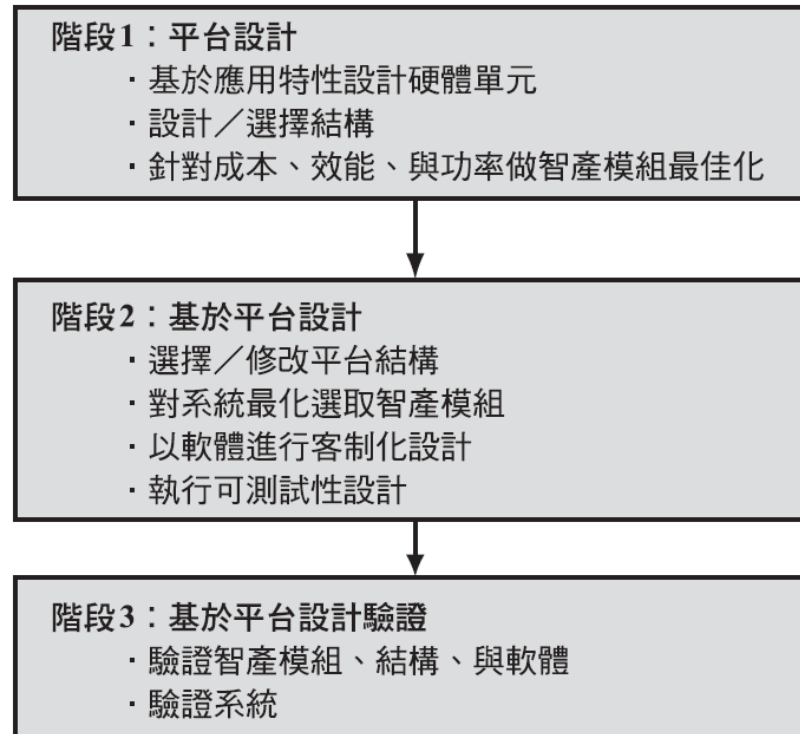


圖 3.6 基於平台設計階段

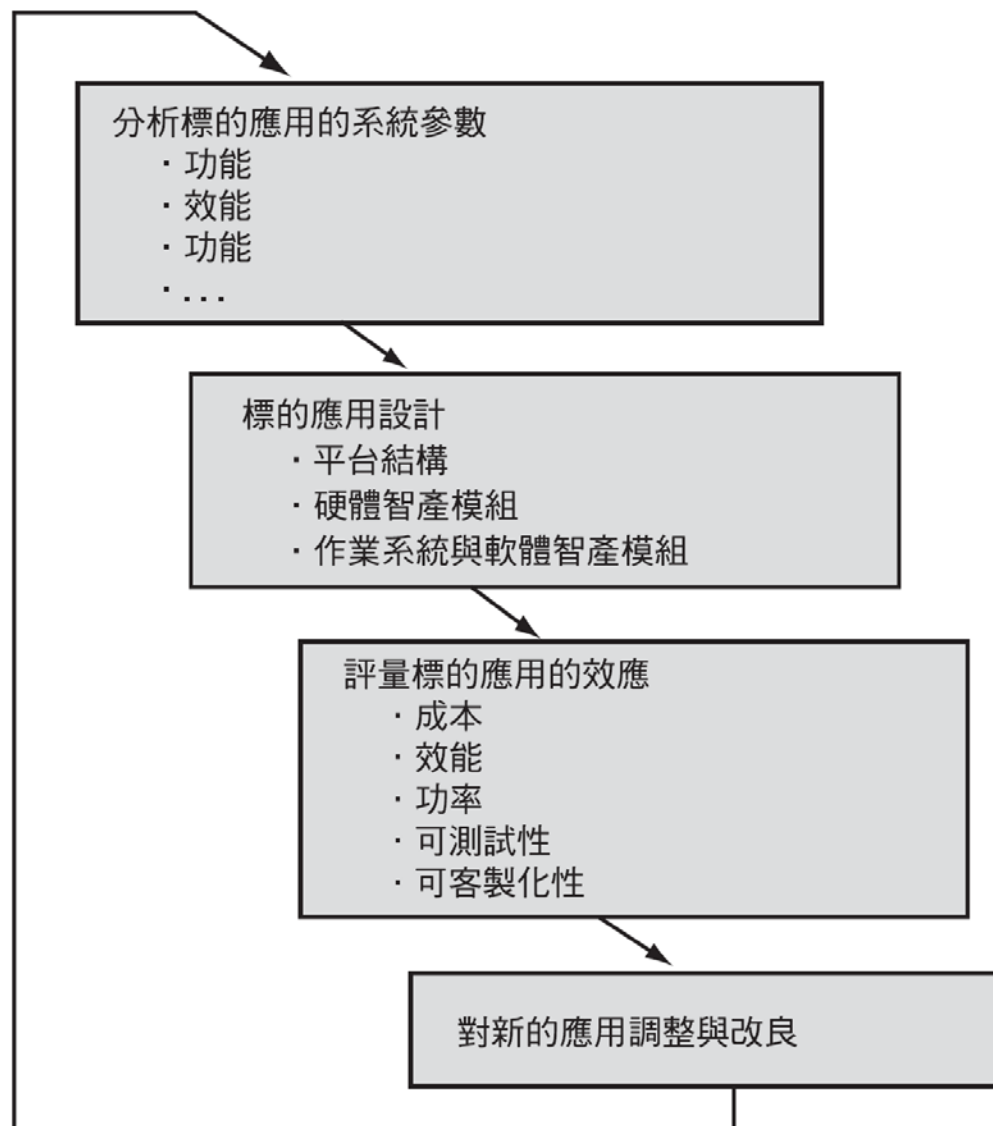


圖 3.7 平台設計步驟

- 在**基於平台設計**的路徑上，系統晶片設計工程師需要兩種能力以發展出標的系統。
 - － 必須了解平台的特性與應用的特性與需求，亦需要能將合宜的平台特徵對映至應用的關鍵功能，也可以藉提供新的軟硬體元件修改平台以使其能符合應用需求。
 - － 必須知道如何進行基於平台設計，包括在成本、效能、功率等方面的最佳化折衝，並維持設計與驗證複雜度在可管理範圍。

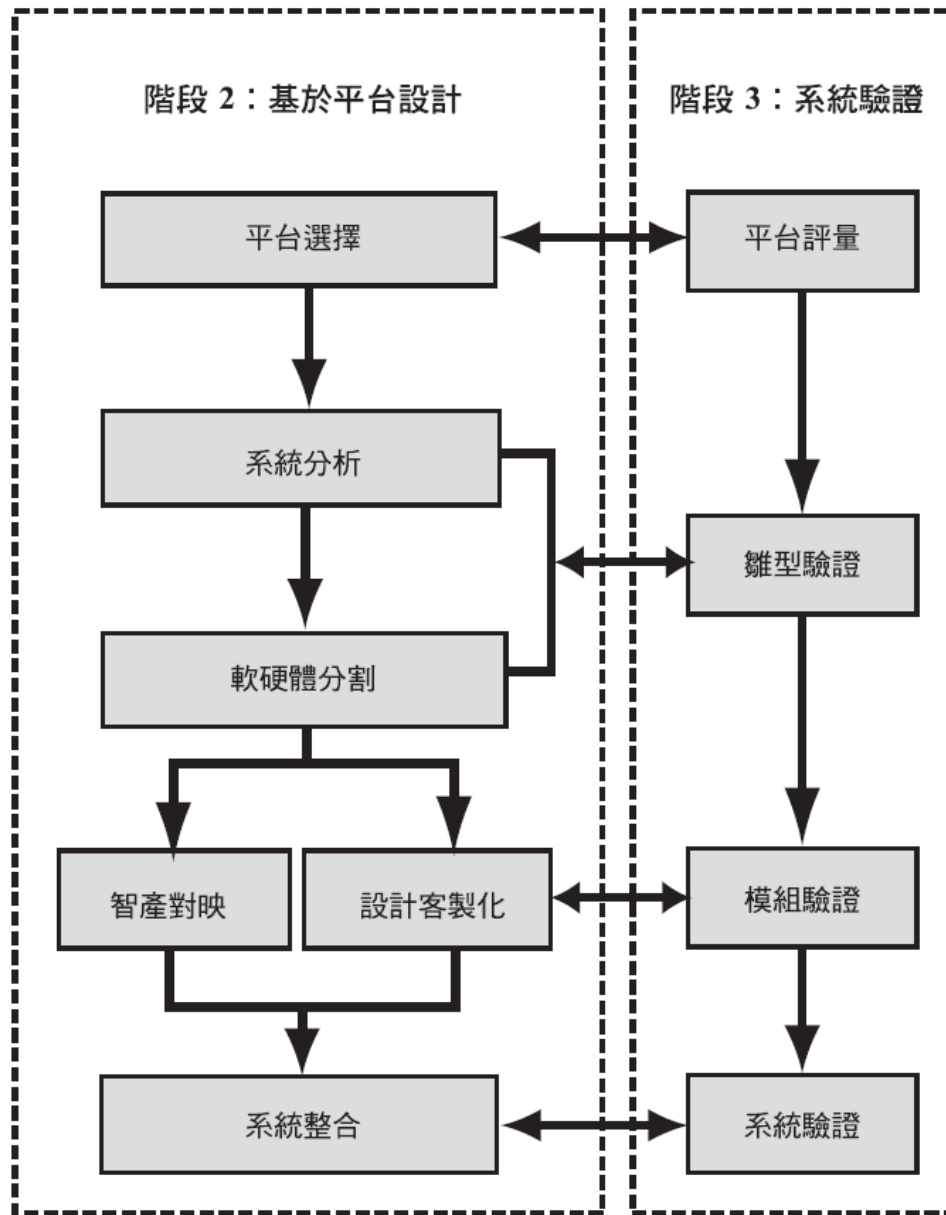


圖 3.8 基於平台設計流程

- 階段 2 的設計工作組成說明如下：

- 1. 平台選擇

- －再用一組存在的平台
 - －修改一組過去的平台
 - －創作一組新的平台

- 2. 系統分析

- －時間關鍵功能
 - Ex. Data Access
 - －複雜度關鍵功能
 - Ex. Control
 - －非關鍵功能

3. 軟硬體分割

- 時間關鍵功能
- 複雜、可改變（複雜度關鍵）與非時間關鍵的功能

4. 智產模組對映

- IP Selection

5. 訂製設計

6. 系統整合

- 在基於平台設計程序中，設計工程師可進行**基於平台設計驗證**以儘早找出並更正設計錯誤。
- 對於雛形驗證，當設計工程師規劃出一個系統雛形，其包含功能模組的分解與其上層結構，此可以被敘述為**雛形模型 (prototype model)**，並使用其驗證總成本、系統功能、效能與功率等需求。

- 智產模組在提供時已經驗證，然而設計工程師仍需要對這些智產模組做重要之**使用案例 (use case)** 的再次驗證，以照應並處置有可能存在的智產模組提供前所做驗證的涵蓋率失誤。
- 在系統整合時，可進行系統驗證，首先分別驗證整合的軟體子系統與硬體子系統，當兩種子系統整合時，再進行**軟硬體協同驗證 (hardware-software co-verification)**，此可以協同模擬 (**co-simulation**) 或協同仿真 (**co-emulation**) 技術達成。

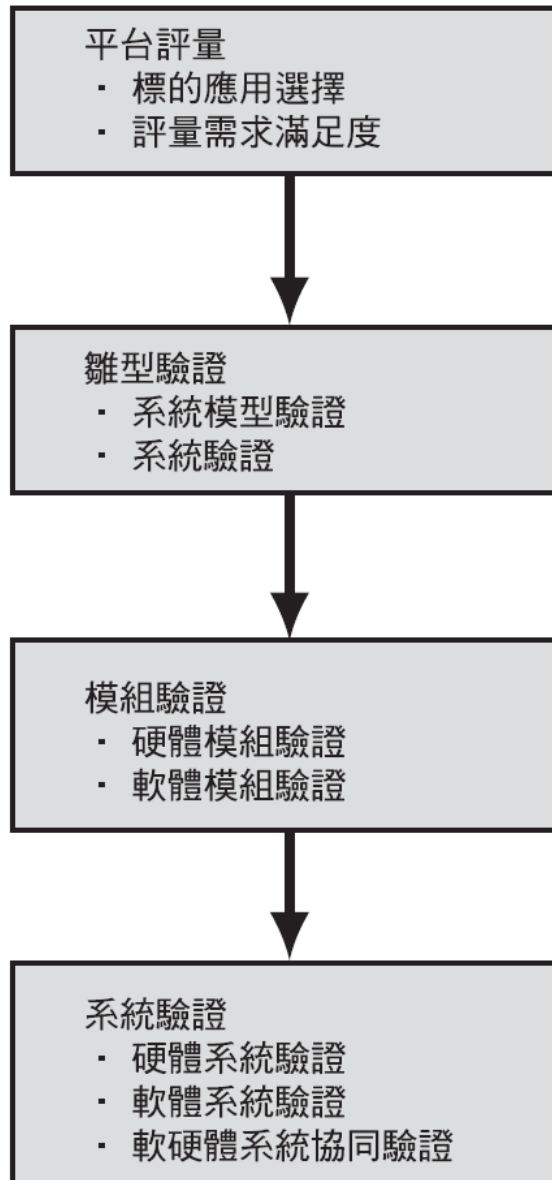


圖 3.9 設計驗證步驟

3.4 基於平台設計的技術

- 對於可以對映到硬體或軟體形式的功能，設計工程師考慮將其對映到以處理器與記憶體執行的軟體設計與以加速器智產模組執行的硬體設計，此種比較提供資訊給**軟硬體分割**工作做為參考。

- 針對高度運用的計算密集與資料存取密集的功能，可使用**硬體加速器**，以硬體功能實作可形成緊密、快速與功率有效性的實作。
- 通常使用軟體編碼執行將工作派遣到硬體加速器上執行相關功能，設計工程師可以將此類控制程式寫為**多執行緒 (multithreaded)** 流程，讓某些執行緒 (**thread**) 將對映到硬體的功能派遣到硬體加速器執行，並管理與其之間的同步工作。

- 當多個功能被對映到軟體程式而使用相同**處理器**時，設計工程師可以選擇合適的處理器以便利程式發展，並考量減低整體執行時間與功率消耗，其可以以兩個步驟選擇執行這些功能的處理器。

- 在系統晶片設計上，**智產模組**的再用為一增進設計生產力的主要技術。
- 當功能被對映至硬體實現形式時，此種尚未選取實作模組的硬體功能被稱之為**虛擬元件** (virtual component)。

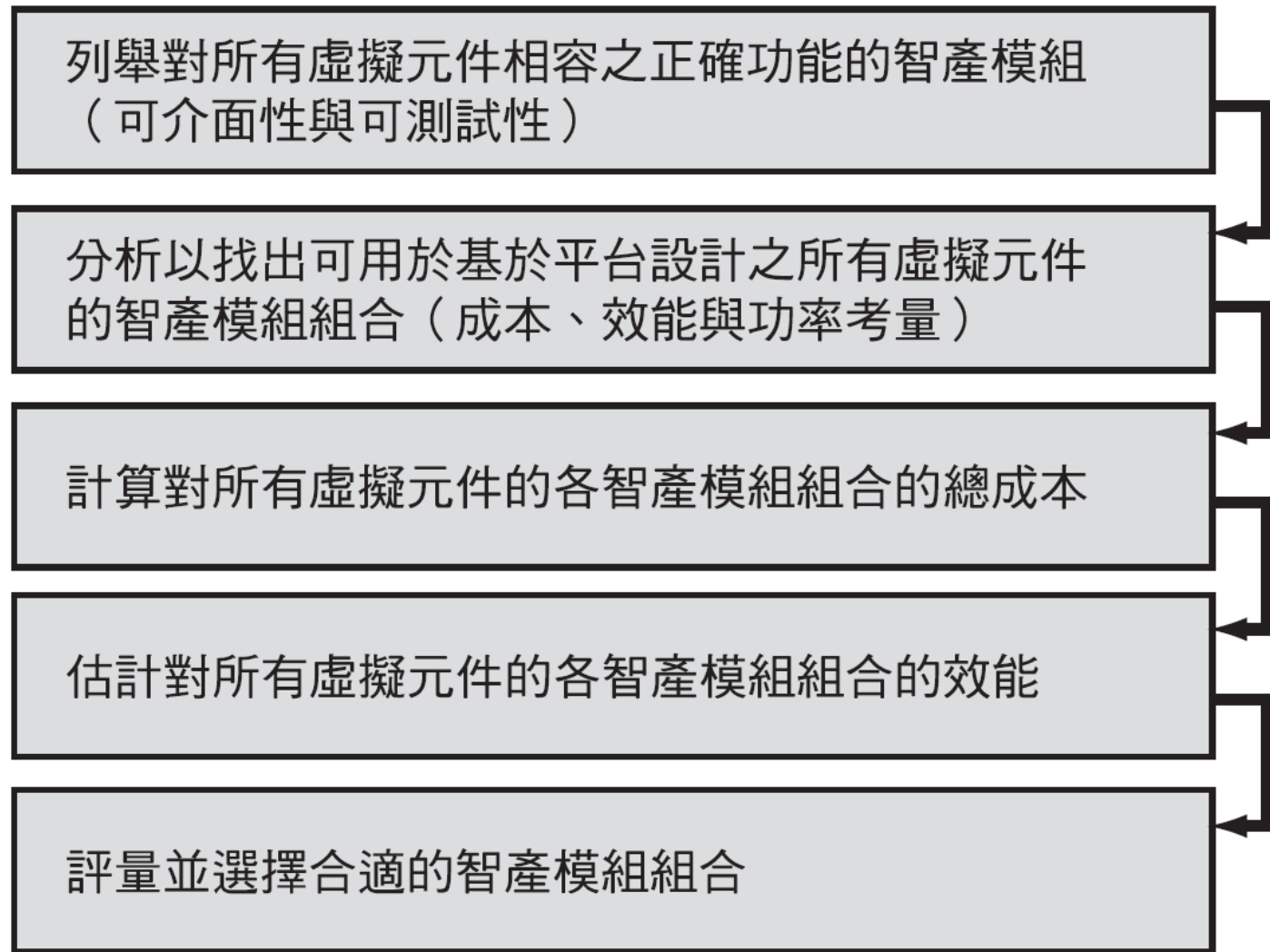


圖 3.10 硬體智產選擇步驟

- 為了由設計工程師自身發展**傳承硬體** (legacy hardware) 智產模組，必須確定一些條件能成立。
- 發展一個智產模組的主要功能與發展一個**VLSI**硬體模組的主要功能是相近似的，共有四種**硬體介面**方式：
 - － 直接連接至目的模組
 - － 透過介面通道作間接連接
 - － 連接至處理器的協同處理器介面
 - － 連接為輸入與輸出裝置

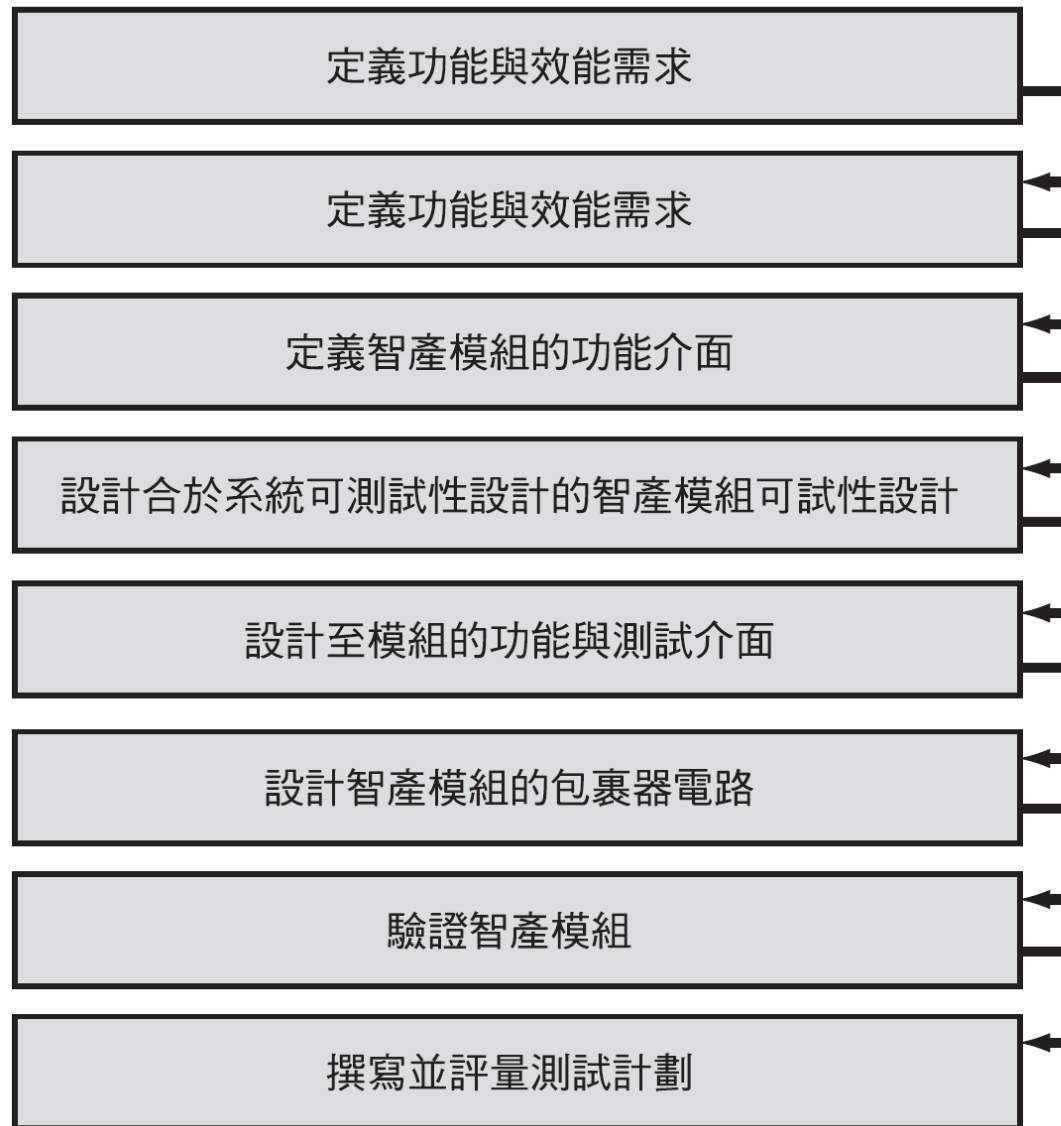


圖 3.11 硬體智產設計步驟

- 現今已有許多**軟體智產模組**可以被運用，其可分為三種類型：
 - 第一類為嵌入式作業系統 (operating systems, OS) 與即時作業系統 (real-time operating systems, RTOS)。
 - 第二類包括基本輸出入系統 (BIOS) 與驅動程式 (driver)，此類功能處理硬體的組態。
 - 第三類為應用層的智產模組，其提供應用層可再用的軟體功能。

- 當沒有合適的軟體智產模組可以用於映射至軟體的功能時，可以設計所需之**傳承軟體 (legacy software)** 智產模組。
- 當將訂製的非智產軟體形成多執行緒時，每一執行緒對應至一組軟體功能的**使用案例 (use case)**。

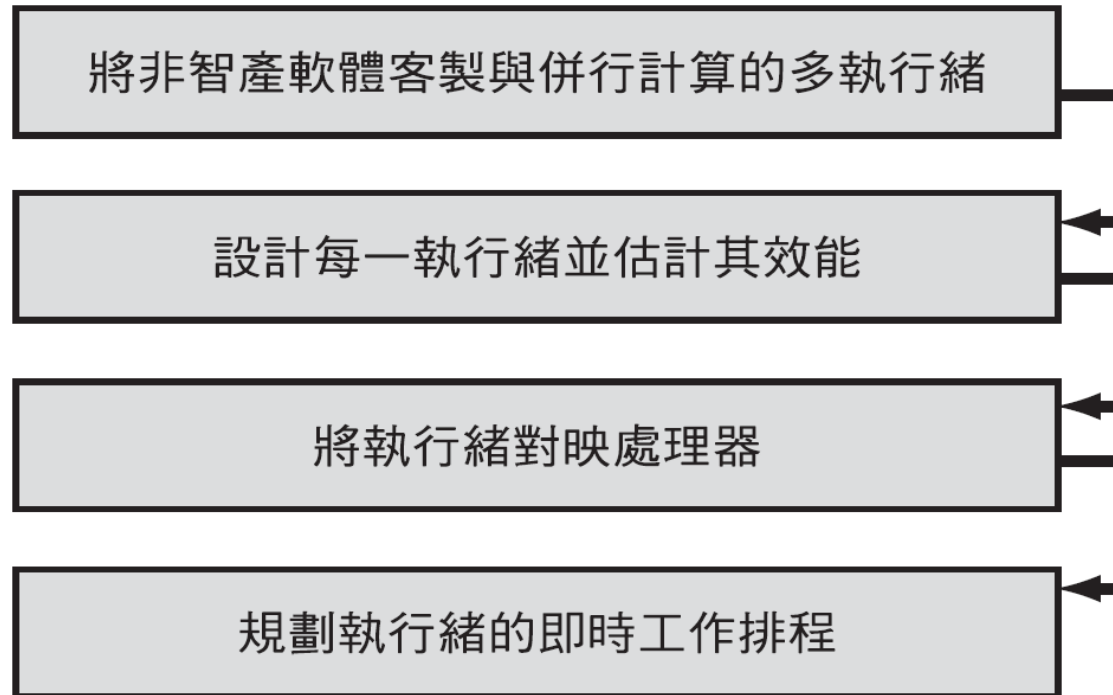


圖 3.12 軟體智產設計步驟

- 記憶體模組類型可以分為三類：
 1. 晶片內記憶體
 2. 緩衝記憶體
 3. 晶片外記憶體

- 外界**RAM**模組可以以四種方式使用：
 - 直接存取外界記憶體
 - 自動化地在外界記憶體與快取記憶體間互傳資料
 - 手動的方式在外界記憶體與晶片內記憶體間互傳資料
 - 透過**直接記憶體存取 (DMA)** 進行有安排的晶片內記憶體與外界記憶體間的資料傳遞

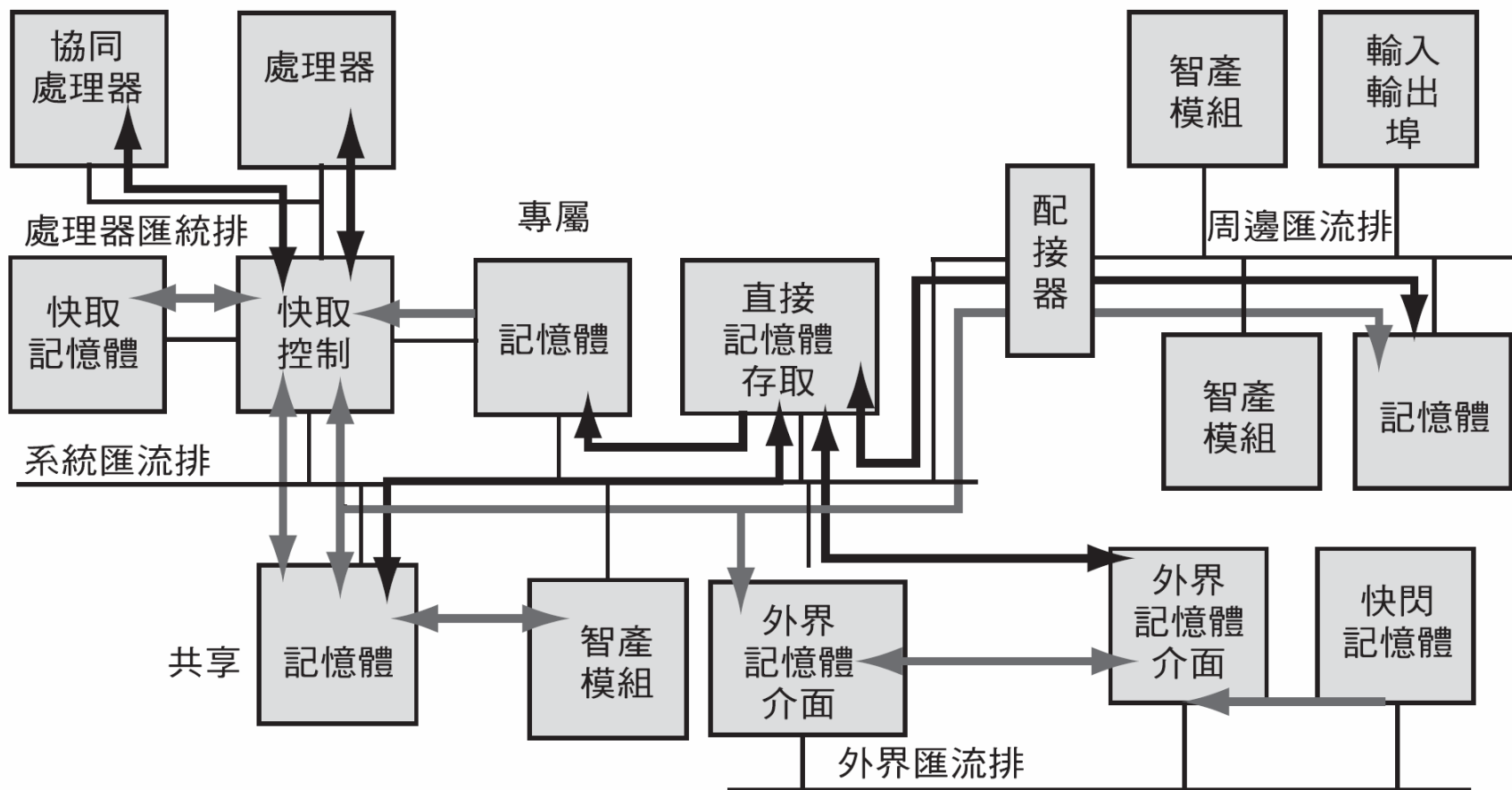


圖 3.13 系統晶片記憶系統設計範例

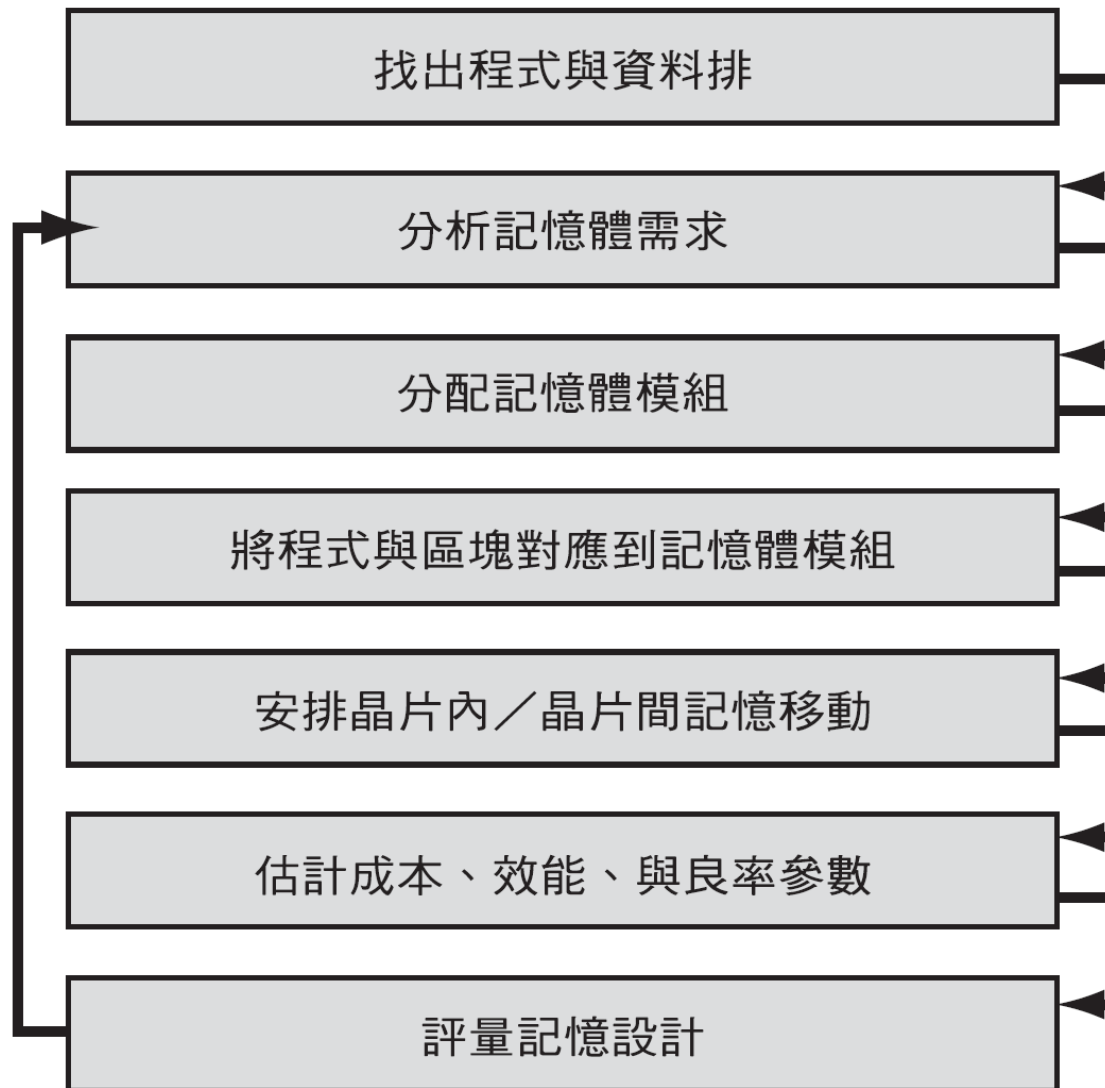


圖 3.14 系統晶片記憶系統設計步驟

- 有兩種方式可以評估記憶系統效能：
 1. 記憶體通量分析
 2. 記憶模擬

表 3-1 記憶體區塊特性

區塊	頻寬	大小
程式	40	1250
A	10	500
B	5	2000

表 3-2 記憶體模組特性

#	類型	大小	成本	頻寬
a	SRAM	500	12	70
b	SRAM	1000	20	50
c	DRAM	2000	30	30
d	DRAM	4000	55	20
e	解壓縮器		3	

- 如圖3.15所示，有五種記憶系統設計：

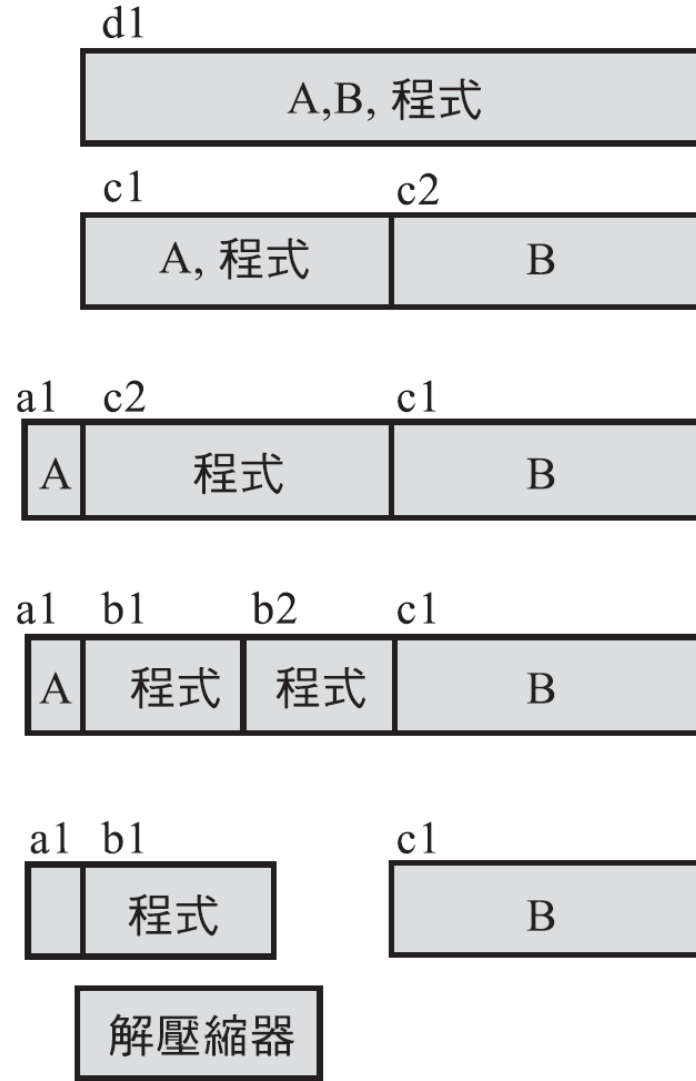


圖 3.15 記憶體分配與指定的選擇

表 3-3 記憶系統設計的頻寬評量

#	模組分配	A	B	程式	通量	成本	解
1	d1	d	d	d	d1: $20 < 85 \times 1.25$	55	無
2	c1, c2	c1	c2	c1	c1: $30 < 80 \times 1.25$, c2: $30 \geq 5 \times 1.25$	60	無
3	a1, c1, c2	a1	c1	c2	a1: $70 \geq 40 \times 1.25$, c1: $30 \geq 5 \times 1.25$, c2 : $30 < 40 \times 1.25$	72	無
4	a1, b1, b2, c1	a1	c1	b1, b2	a1: $70 \geq 40 \times 1.25$, c1: $30 \geq 5 \times 1.25$, b1,b2 : $50 \times 2 \geq 40 \times 1.25$	82	有
5	a1, b1, c1 解壓縮器	a1	c1	b1	a1: $70 \geq 40 \times 1.25$, c1: $30 \geq 5 \times 1.25$, b1 : $50 \geq 40 \times 1.25$ (壓縮 1250 為 900)	65	有

- 為了在模組間傳輸指令與資料，系統需要晶片內**通訊系統**，多種通訊資源可以被使用來建立一組訂製的通訊系統。

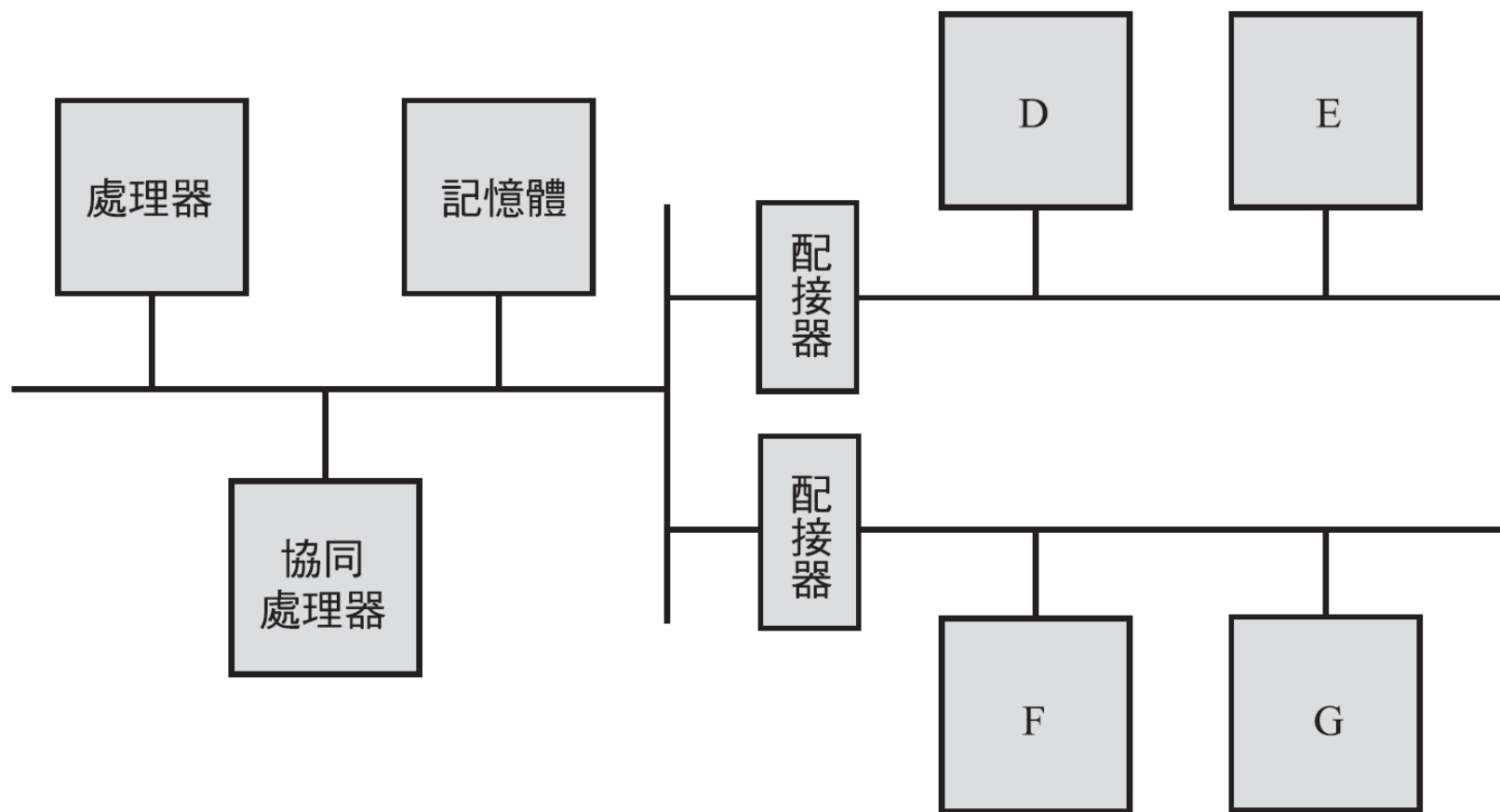


圖 3.16 通訊結構範例

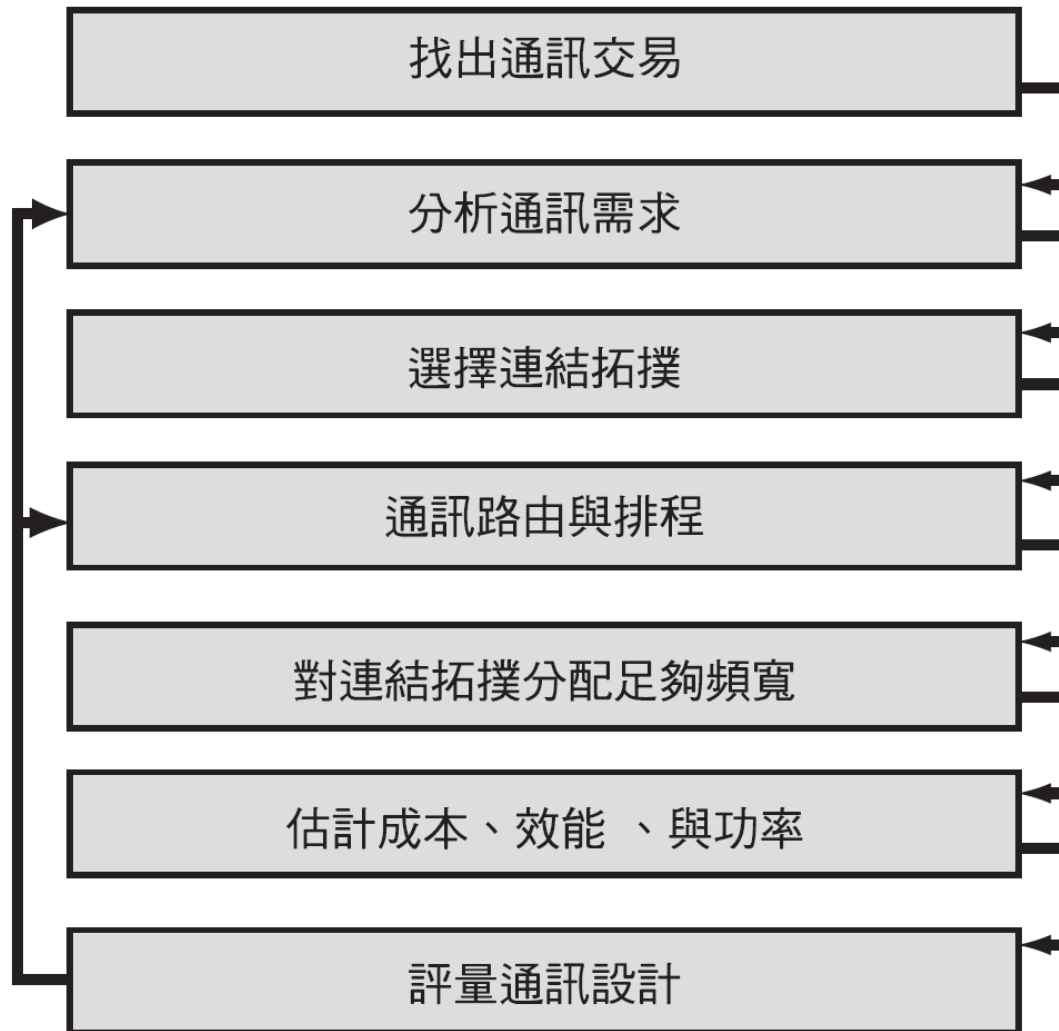


圖 3.17 系統晶片通訊系統設計步驟

- 在通訊系統設計程序中，可以運用數種設計技術：
 1. 為了決定通訊要求 (communication request) 的**通訊潛伏期**，設計工程師必須根據良好的晶片樓層規畫 (chip floor plan) 設計一組合適的連結拓撲。
 2. 對於時間關鍵的通訊要求，其可以分配專屬的**通訊鍵結**，以增加關鍵點的頻寬。當傳輸通過共享的通訊鍵結（**匯流排**），其亦可對此類要求指定高優先權 (priority)，共享的鍵結亦可以用來傳輸非時間關鍵的通訊要求。

3. 長訊息的通訊潛伏期 (communication latency) 可以使用**蟲孔傳輸** (wormhole transmission) 與管線降低，而短訊息的最大通訊潛伏期與實體通道運用率可以使用透過**虛擬通道** (virtual channel) 的時間多工獲得改進。
4. 設計工程師可以藉由適當分配通訊頻寬，並且對通訊網路上的訊息做有效路由與排程，可降低通訊瓶頸與通訊成本。舉例來說，藉由對長潛伏期的訊息做**分割通訊交易** (splitting communication transaction)，可以增進通訊的通量。
5. 運用匯流排與交換器等連結通道，實作硬體元件間共享的通訊設計，滿足其間的通訊要求。

- 設計出通訊系統設計後，設計工程師應評量其成本效益是否符合所有系統需求，有兩個方式可以用來評估通訊系統的效能：
 1. 通訊通量分析
 2. 通訊模擬

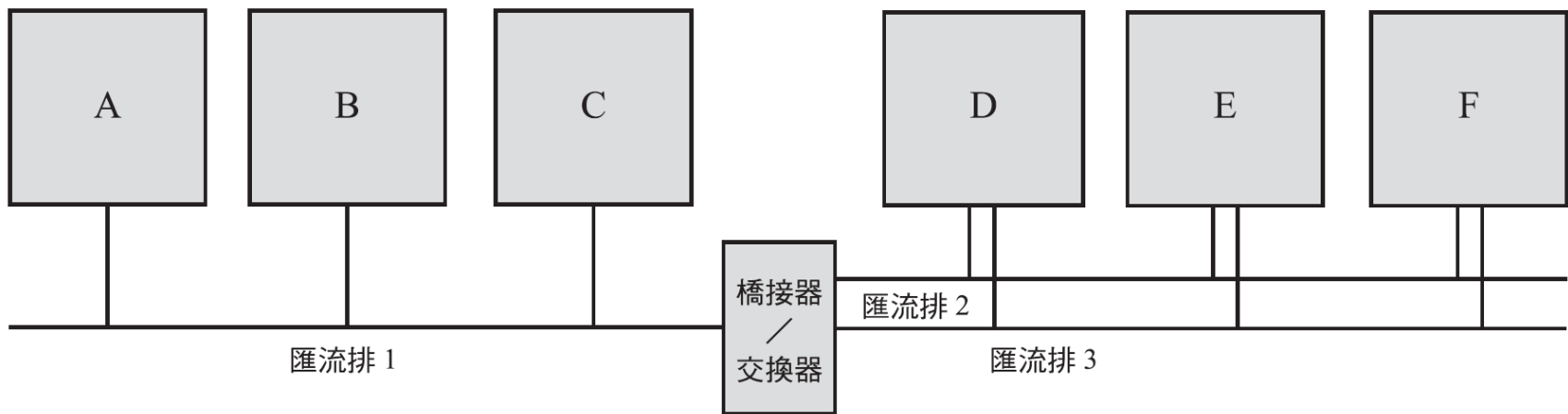


圖 3.18 通訊設計結構範例

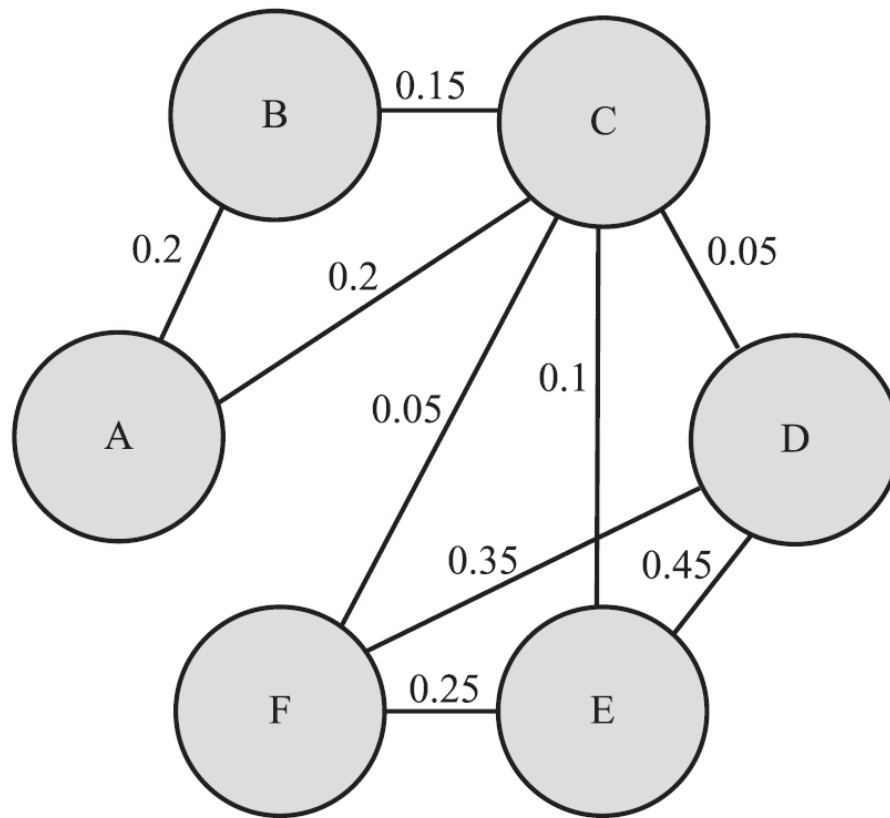


圖 3.19 通訊設計頻寬需求範例

- 針對此通訊結構，給定一組在此三個匯流排上的路由方式，可以以下列方式評量：
 1. 在匯流排 1 上
 2. 在匯流排 2 上
 3. 在匯流排 3 上

- 各個硬體與軟體智產模組被選出或是設計出來，同時處理器亦被選出，並將軟體對映至其上執行時，設計工程師可以使用這些元件進行此系統設計的**系統整合**。

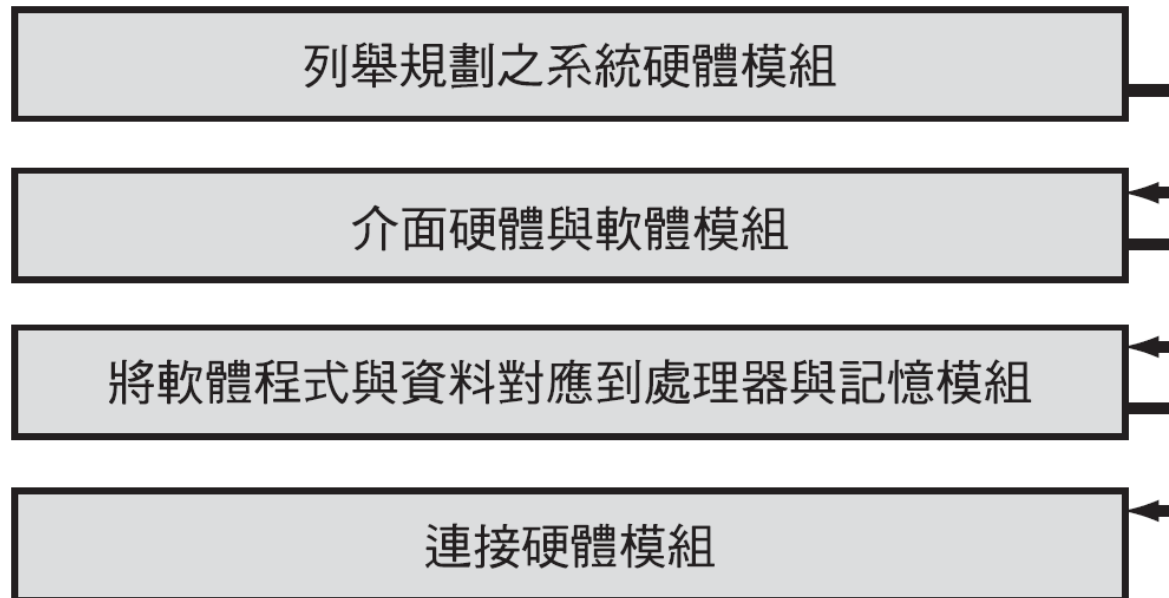


圖 3.20 系統整合步驟

- 對於軟體智產模組驅動硬體智產模組的使用，設計工程師可以使用**協同處理器介面 (co-processor interfacing)** 或**記憶體對映輸入與輸出介面 (memory-mapped I/O interfacing)** 方式。
- 對於硬體智產模組驅動軟體智產模組的使用，設計工程師可以使用**中斷處理 (interrupt processing)** 與**輪詢輸入與輸出 (polled I/O)** 等介面技術。

- **基於平台設計驗證**包括平台評量、雛型驗證、模組驗證與系統驗證。
- 設計工程師可以為標的應用作**系統雛型驗證**。

- 硬體智產模組為一功能模組，如其細節技術內容可以全部（或大部分）可取用，則可以使用傳統硬體驗證的方法即**白箱測試 (white box testing)** 加以驗證，不然，只能使用**黑箱測試 (black box testing)** 的方法驗證。
- 軟體模組也需使用軟體測試的方法驗證，對於其效能與功率驗證，通常使用透過**指令集模擬 (instruction set simulation, ISS)** 模擬在標的處理器上執行的狀況加以估計。

- 整合驗證主要檢查整合後之軟硬體組合是否能正確運作，其包含硬體介面驗證、軟體整合驗證與**軟硬體協同驗證 (hardware-software co-verification)**。
- 軟硬體整合的驗證為一主要的系統晶片的驗證工作，有兩種軟硬體協同驗證方法
 - － **軟硬體協同模擬 (hardware-software co-simulation)**
 - － **軟硬體協同仿真 (hardware-software co-emulation)**。

- 設計工程師可以使用包含嵌入式處理器、嵌入式記憶體、硬體智產模組巨集、與訂製硬體邏輯的**可程式邏輯閘 (FPGA)** 模組等之系統晶片發展板做為協同仿真的平台，其可將硬體與軟體模組對映到協同仿真的平台上，在對系統晶片設計進行協同仿真。
- 在協同仿真過程，仿真的速度較快，因而可以驗證較多的測試案例，其並可介面並與應用環境直接互動，可對此系統設計進行**在線現場測試 (on-line field test)**。

3.5 基於平台設計的分析

- 基於平台設計的效能需求可分為兩類，如下將介紹這兩類設計的**效能估計**技術：
 - 非即時 (non-real-time) 需求
 - 即時 (real-time) 需求

- 為了估計一組**非即時並行工作**的效能，設計工程師可進行如圖3.21的效能估計步驟。

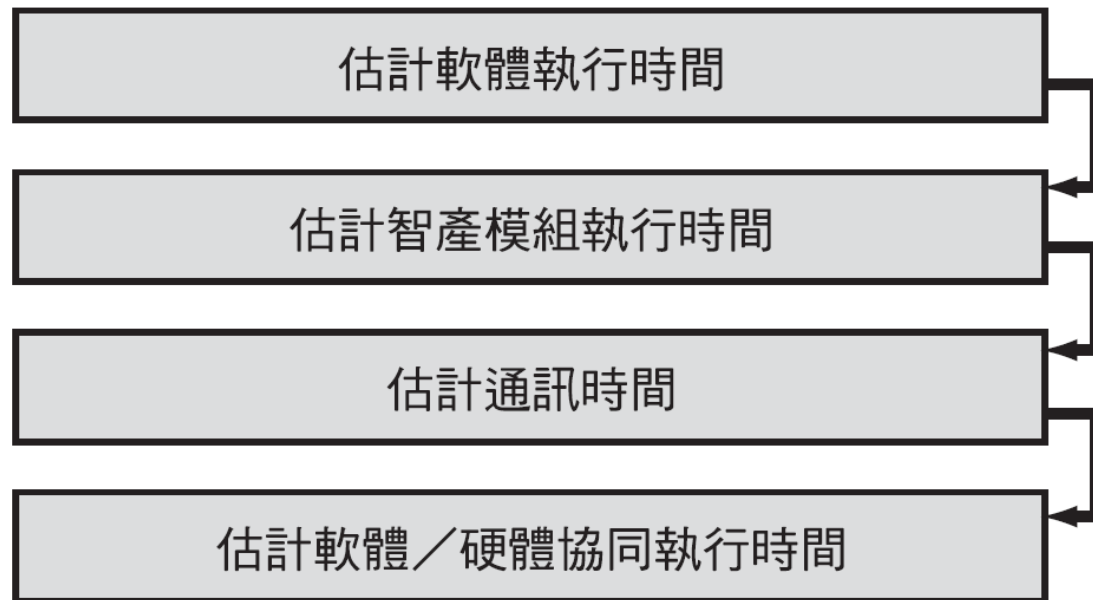


圖 3.21 非即時系統的效能評估步驟

- 整合這些估計資料為整體軟硬體效能估計結果，步驟分別說明如下：
 1. 估計軟體執行時間
 2. 估計智產模組執行時間
 3. 估計通訊時間
 4. 估計軟硬體協同執行時間

- 為了滿足並行工作的即時需求，設計工程師通常採取三種排程技術：
 - 固定循環性排程 (Fixed cyclic scheduling)
 - 嚴格速率排程 (Rate monotonic scheduling)
 - 最早截止期第一排程 (Earliest-deadline-first scheduling)

- 如果並行工作為週期性的發生，設計工程師可以安排這些工作於**固定循環性排程**。
- 如果工作在執行時隨機的產生，設計工程師可以對並行工作採用**嚴格速率排程**或**最早截止期第一排程**，兩種排程均為優先權排程 (priority scheduling) 技術，其允許擁有最高優先權的工作奪取目前工作的執行權。

- 即時設計的可排程性 (schedulability) 為所有並行工作的時間限制 (timing constraint) 是否均可符合，即時分析 (real-time analysis) 可決定即時設計的可排程性。

- 動態功率消耗的估計技術，共有三種動態
功率估計方法：
 1. 基於機率的估計
 2. 功率估計
 3. 功率模擬

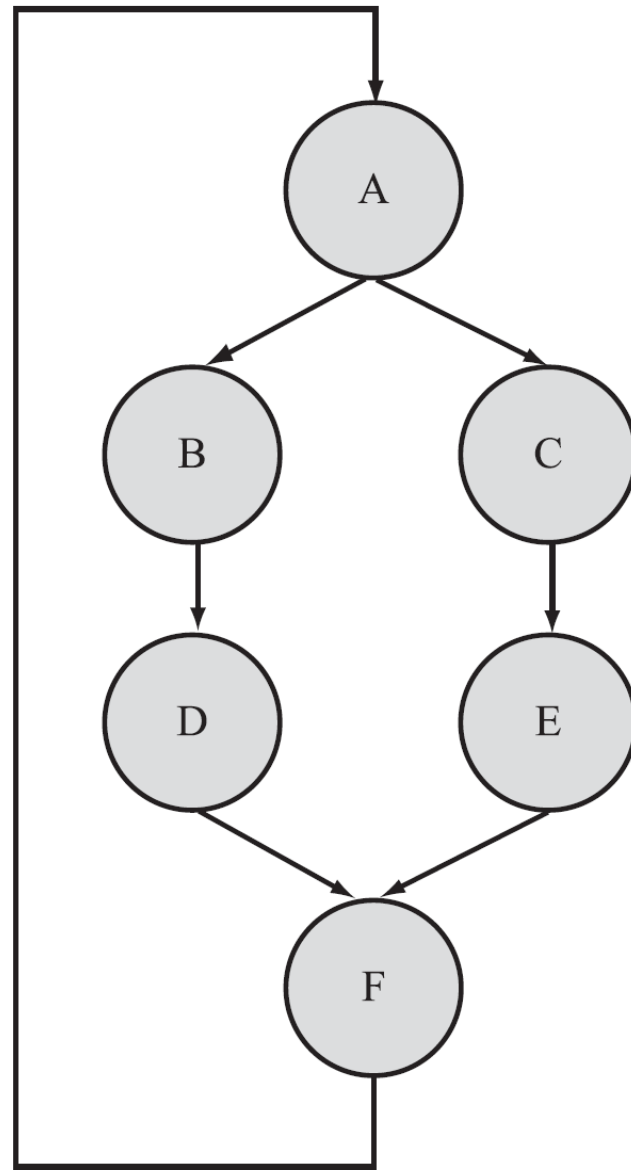


圖 3.22 系統設計工作圖範例

表 3-4 工作對映與模組成本

工作對映

工作	模組	時間	功率
A	P1	200	50
A	P2	300	40
B	P1	1000	240
B	P2	1300	210
B	C1	700	200
C	P1	600	140
C	P2	800	120
C	C2	500	110
D	P1	300	80
D	P2	400	70
E	P1	900	220
E	P2	1000	190
F	O1	200	100
F	P2	300	50

成本列表

模組	面積
P1	300
P2	200
C1	200
C2	150
O1	100
O2	50

表 3-5 系統設計的評估結果

工作對映

模組分配	A	B	C	D	E	F	面積	時間	功率
P1, O1	P1	P1	P1	P1	P1	O1	400	3200	830
P1, O2	P1	P1	P1	P1	P1	O2	350	3300	780
P2, O1	P2	P2	P2	P2	P2	O1	300	4100	730
P2, O2	P2	P2	P2	P2	P2	O2	250	4200	680
P1, C1, O1	P1	C1	P1	P1	C1	O1	600	1800	750
P1, C1, O2	P1	C1	P1	P1	C1	O2	550	1900	700
P2, C1, O1	P2	C1	P2	P2	C1	O1	500	2000	710
P2, C1, O2	P2	C1	P2	P2	C1	O2	450	2100	660
P1, C1, C2, O1	P1	C1	C1	P1	C1	O1	750	1800	720
P1, C1, C2, O2	P1	C1	C1	P1	C1	O2	700	1900	670
P2, C1, C2, O1	P2	C1	C2	P2	C1	O1	650	1900	700
P2, C1, C2, O2	P2	C1	C2	P2	C1	O2	600	2000	650

3.6 基於平台設計的結構範例

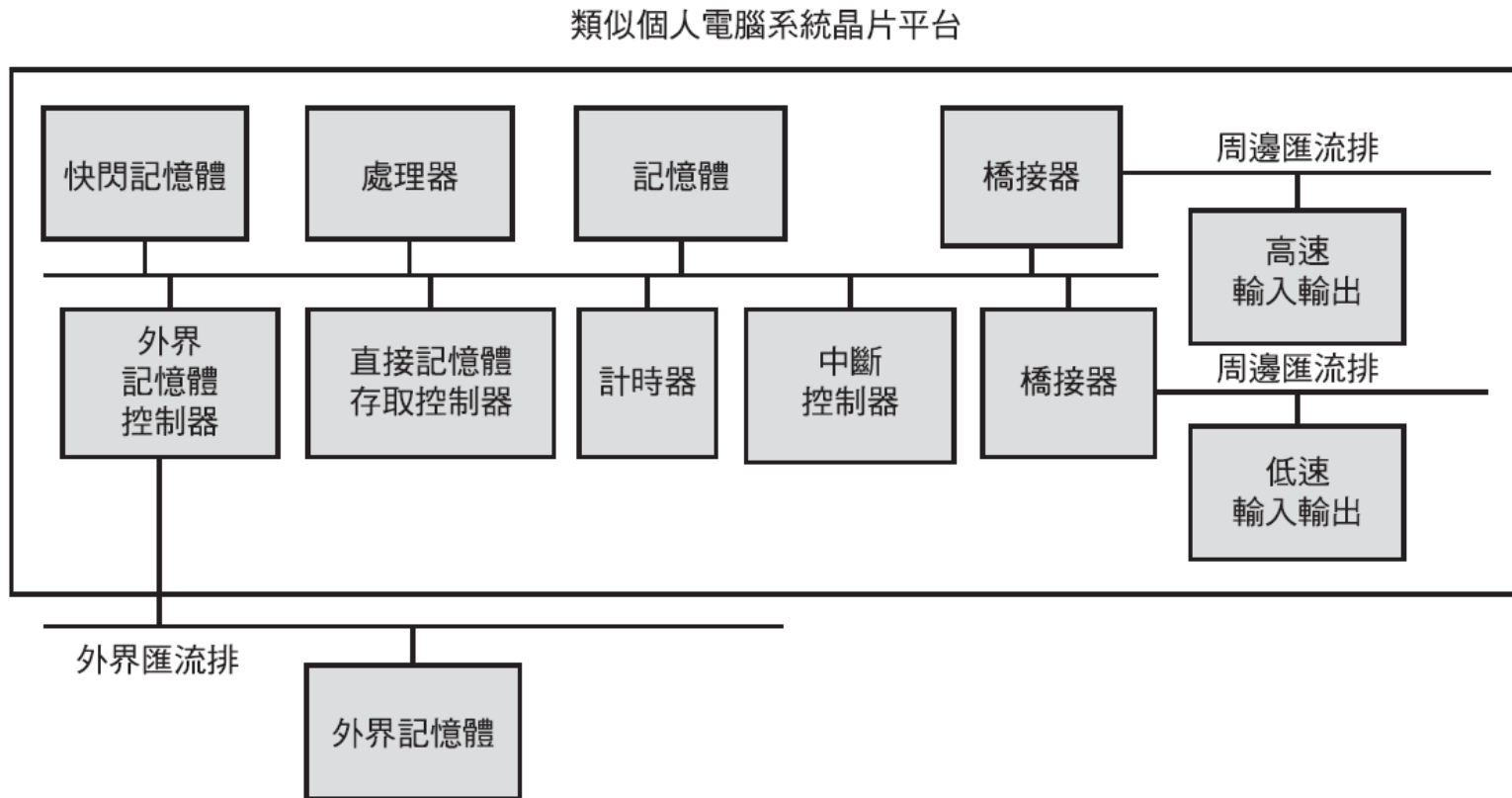


圖 3.23 類似個人電腦平台

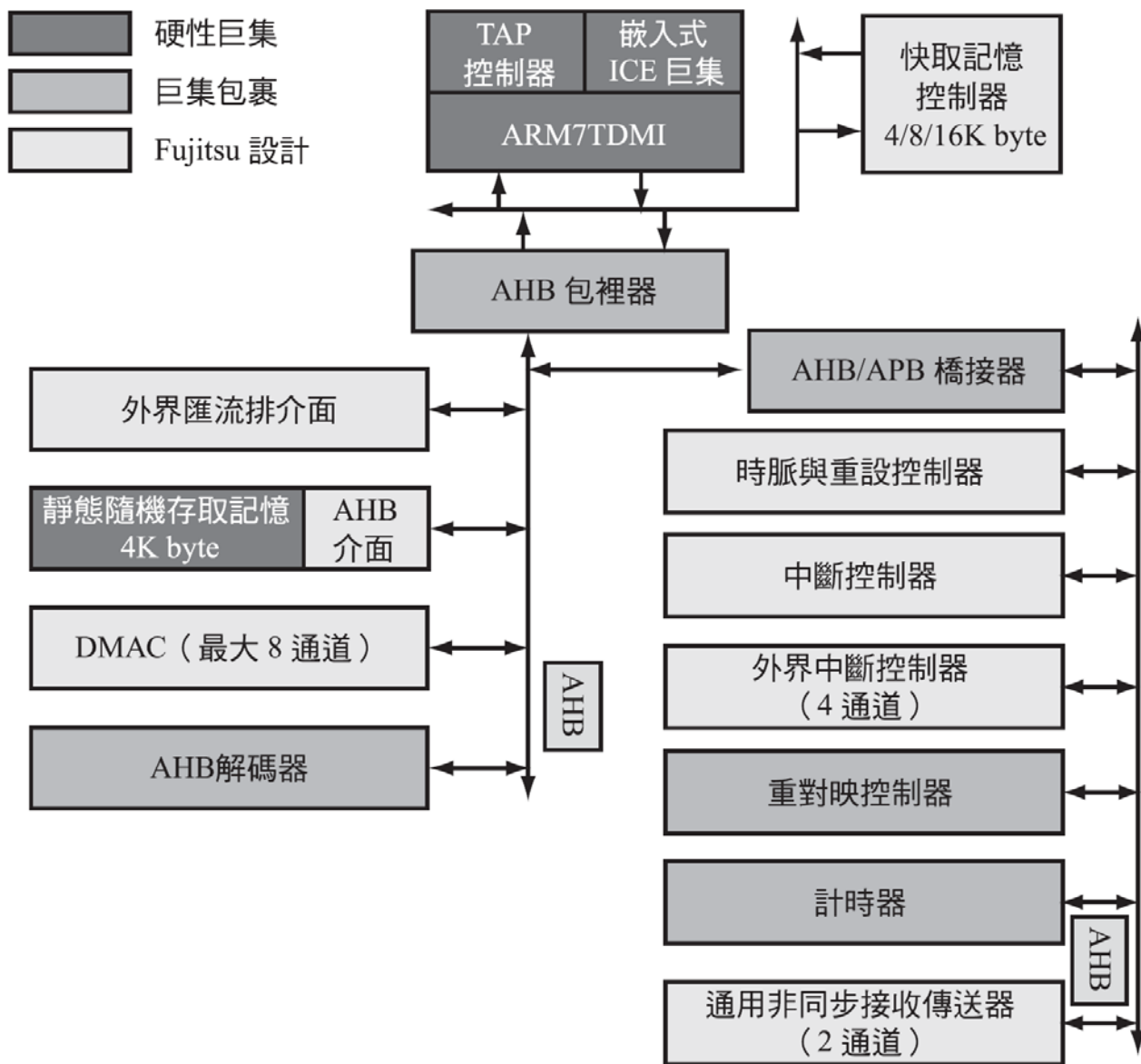


圖 3.24 基於 ARM 的平台

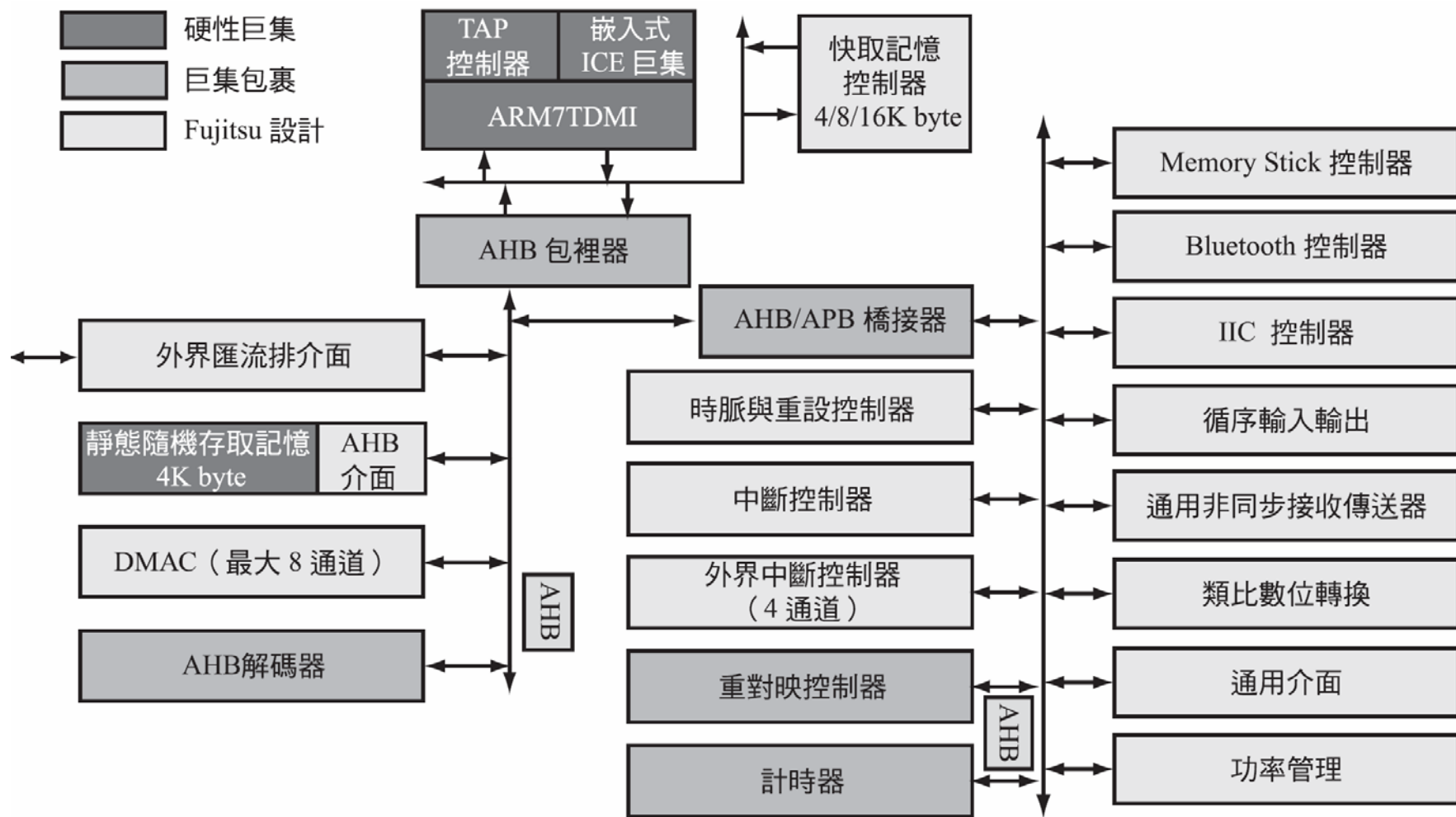


圖 3.25 基於 ARM 平台的基於平台設計

- 要選擇自訂能力的處理器時，可以使用**組態處理器** (configurable processor)，一組 Tensilica Xtensa 可組態處理器提供如下主要組態特徵：
 - － 基礎結構
 - － 可組態功能
 - － 選取性功能
 - － 可選擇與可組態功能
 - － 進階設計工程師定義功能

Xtensa LX 結構

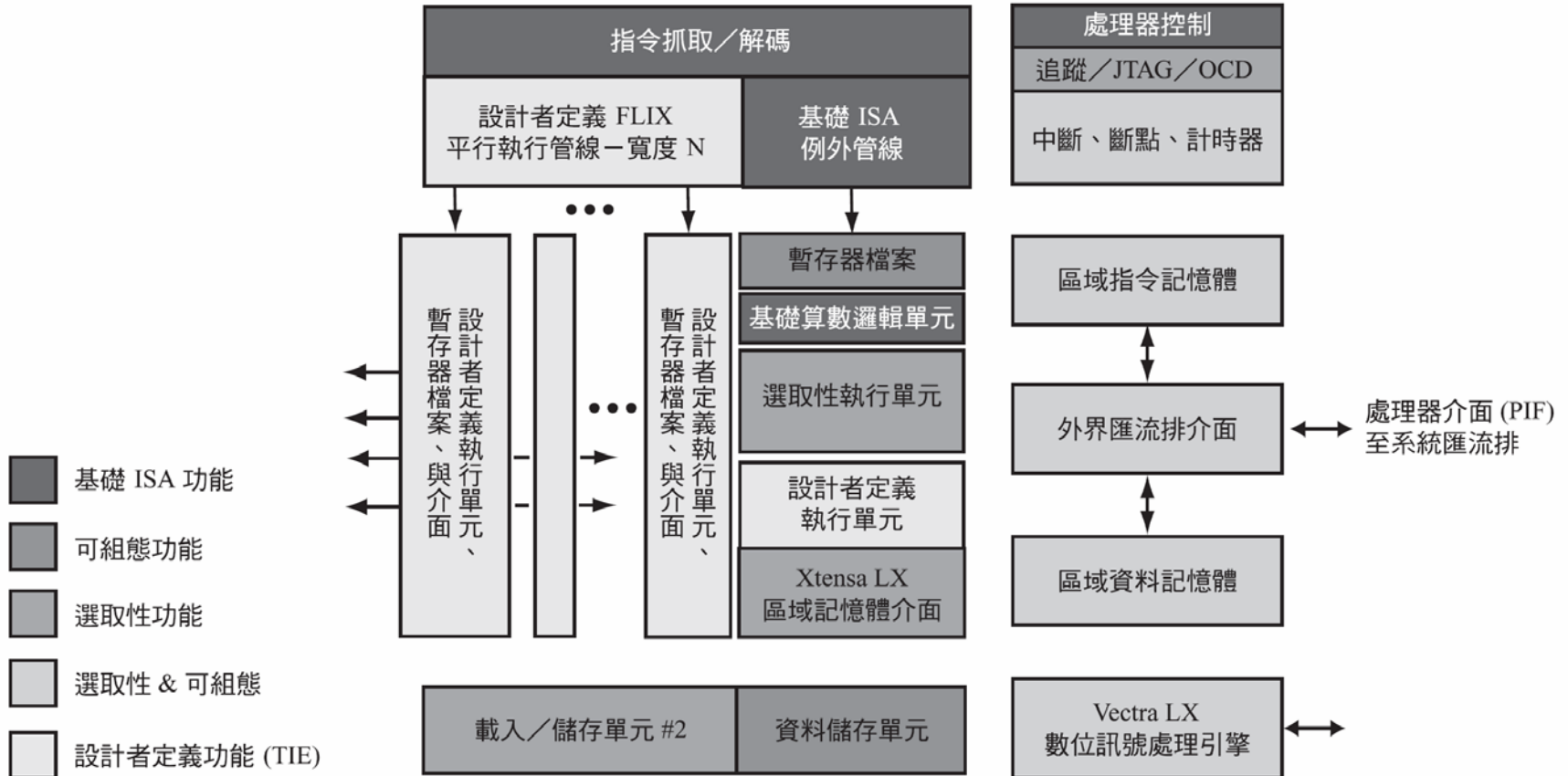


圖 3.26 可組態處理器

- 一組可參數化**數位相機**平台提供四種數位相機所需功能或使用案例的硬體元件與系統結構：
 - － 相片輸入
 - － 相片顯示
 - － 相片輸出
 - － 相片管理

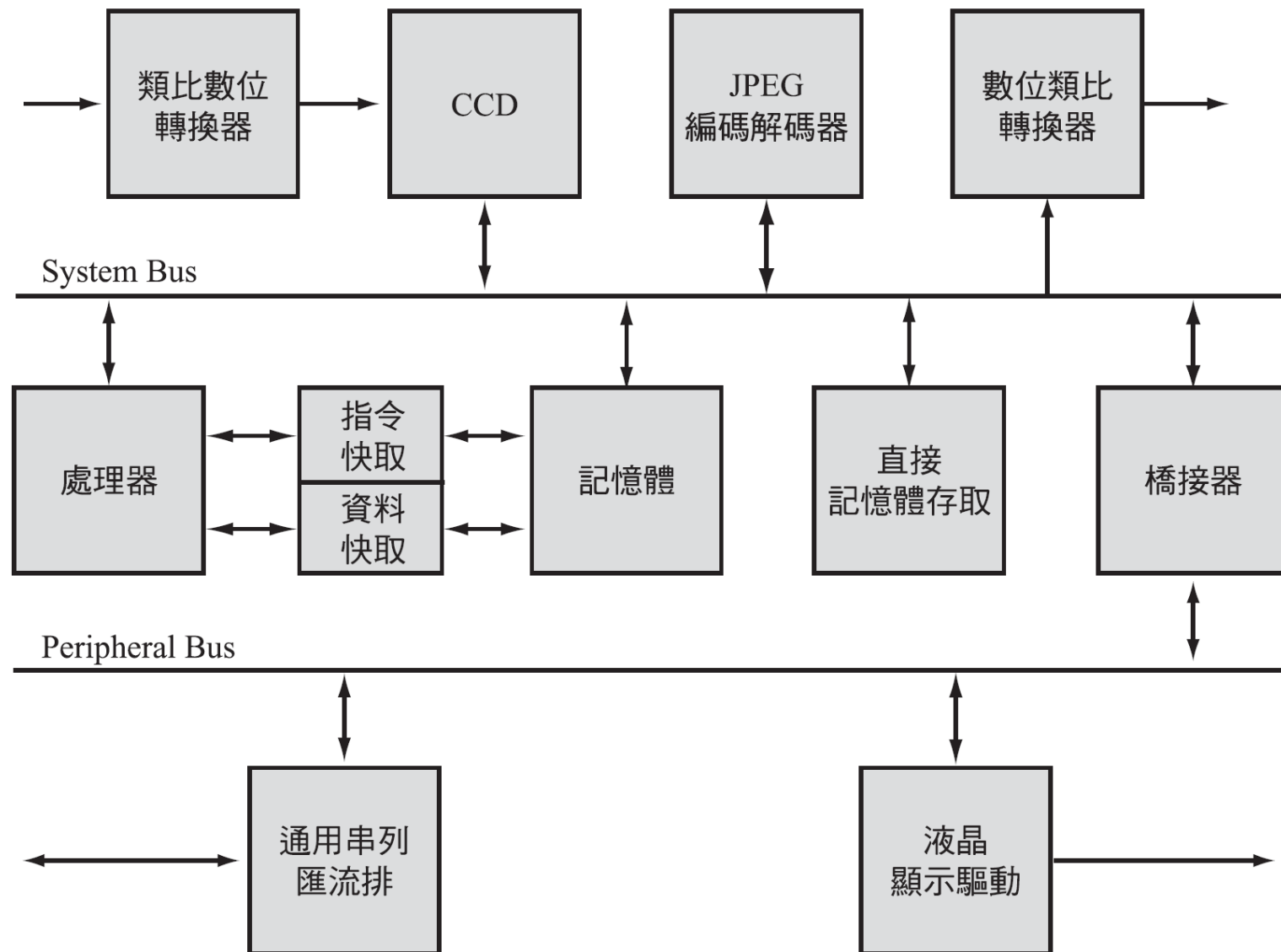


圖 3.27 數位相機平台

- 一組高效能多核心平台可用於多媒體與訊號處理應用，其使用兩組多層**AMBA AHB**匯流排以整合一個主要處理器核心、多個數位訊號處理器核心、與平行記憶體模組陣列。
 - － 此主要處理器核心**OMNIcone**執行純量處理並且控制數位訊號處理運算。
 - － 每一數位訊號處理器核心**ITC RADcore**為一可組態訊號處理協同處理器，其可以執行平行的數位訊號計算，並透過多層 (multi-layer) **AHB**匯流排進行平行的記憶體存取，一個輸入與輸出**EXU**模組提供與記憶體模組陣列與外界記憶體的輸入與輸出、及與其他**RADcore**間的通訊。

- 其連結結構使用兩組匯流排，其一為單層級 **AHB** 匯流排作為從主要處理器至數位訊號協同處理器的控制匯流排。

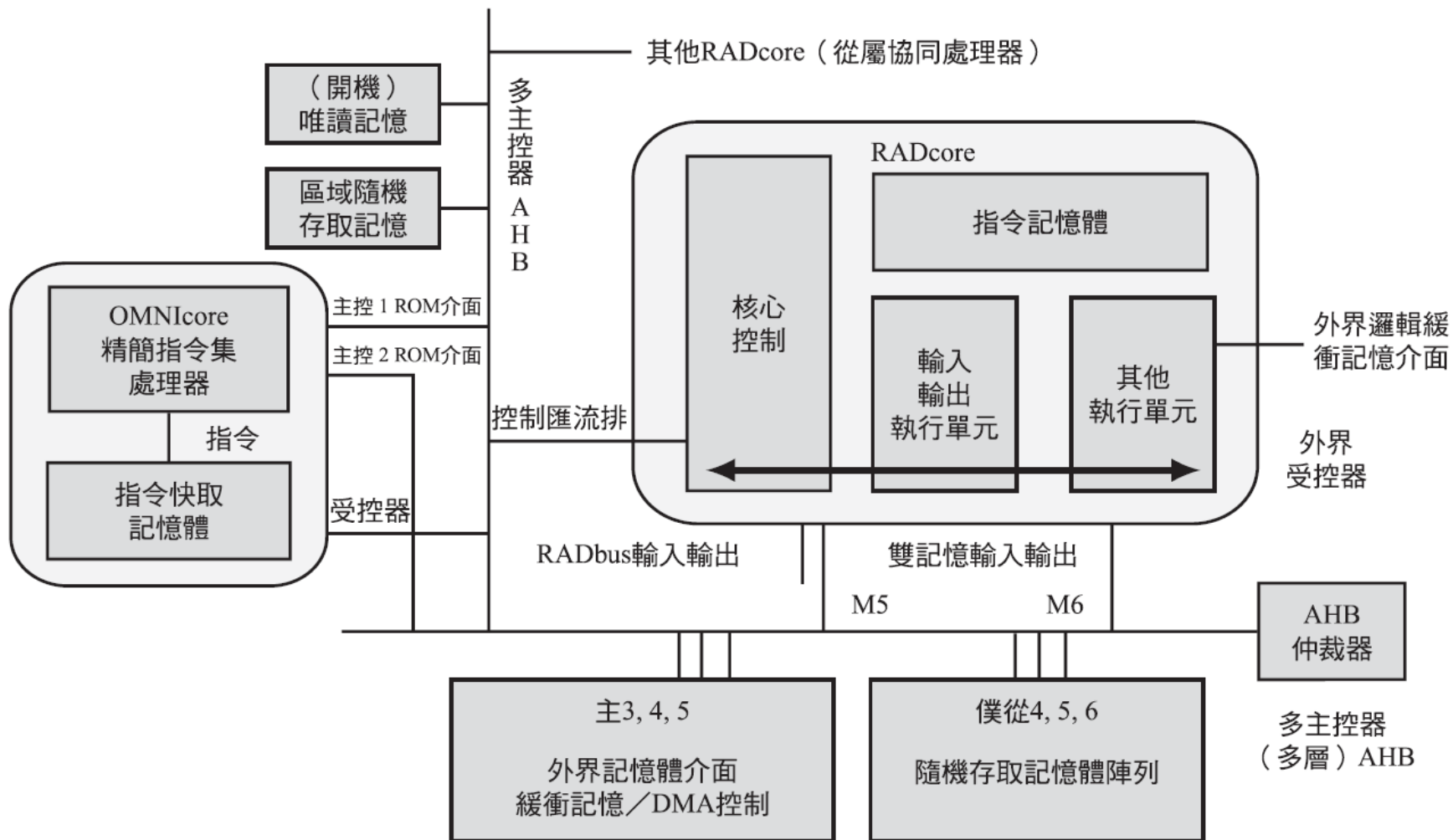


圖 3.28 高能效多核心平台

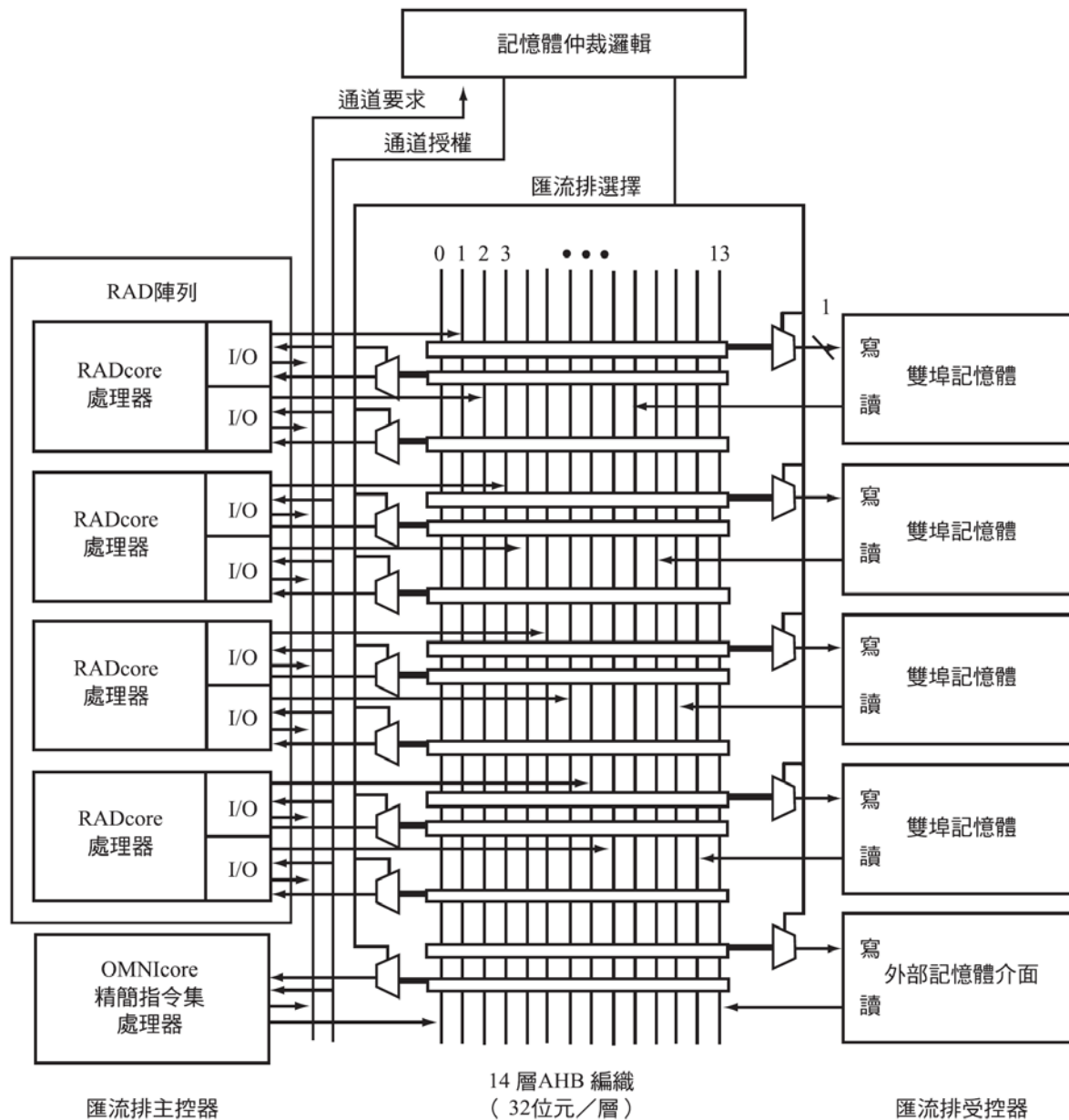


圖 3.29 多層 AHB 匯流排的高頻寬連結電路

3.7 結論

- 基於平台設計方法透過數種設計技術提供適時的複雜系統晶片設計：
 1. 其可以再用可選擇的平台結構、平台中不同的硬體智產模組、連結電路、與軟體智產模組。
 2. 其將系統設計流方法簡化為工作分解、軟硬體分割、與設計對映至可再用與傳承元件上。
 3. 設計工程師因此主要著重於系統層級的設計、分析、最佳化、與驗證工作。
 4. 藉由再用多數的硬體智產模組，標的應用的細節設計可以大部分由軟體完成，此將較為可管理與可維護的。