

# 輔助處理器架構：適用於快速原型開發的嵌入式系統架構

**編者說明** — 雖然輔助處理器架構的數位處理效能和傳輸量已廣為人知，但其還提供嵌入式系統設計人員機會來實作專案管理策略，同時改善開發成本和上市時間。本文特別著重於離散式微控制器 (MCU) 和離散式現場可編程閘陣列 (FPGA) 的合併使用，藉此展示此架構如何帶來高效率的迭代式設計流程。本文將透過前人研究的資料來源、實證發現與案例研究，探討此架構的優點，同時提供堪稱典範的應用。在本文結束時，嵌入式系統設計人員將更加瞭解何時以及如何實作此種多用途硬體架構。

■作者：Noah Madinger

Colorado Electronic Product Design (CEPD)

## 介紹

嵌入式系統設計人員發現自己在設計限制、效能期望、排程以及預算問題的考量之間交戰。確實，甚至是充滿矛盾的現代專案管理口號和詞彙，都進一步凸顯此角色焦慮不安的本質：「快速故障」；「靈活敏捷」；「永不過時」及「具有破壞性」！即便嘗試滿足這些期望所用的技法，可能相當折磨人，但是整個市場都在談論，並且不斷強調其重要性。我們需要的是一種設計方法，能實作不斷進化的迭代程序，而且如同多數的嵌入式系統一樣，都得從硬體架構開始。

輔助處理器架構是一種硬體架構，以微控制器 (MCU) 和現場可編程閘陣列 (FPGA) 技術的合併優勢而著稱，並且可為嵌入式設計人員提供一個流程，讓他們滿足甚至最嚴苛的要求，同時仍具備必要的彈性，以因應已知和未知的挑戰。使用具備迭代調適能力的硬體，設計人員便能展示進度、達到關鍵里程碑，並完全發揮快速原型開發流程的優勢。

這個流程中有關鍵的專案里程碑，每個都有其獨特的價值，可增添到開發工作中。在整篇文章中，這些里程碑將由以下術語指代：使用微控制器進行數位訊號處理里程碑、使用微控制器進行系統管理

里程碑，以及產品部署里程碑。

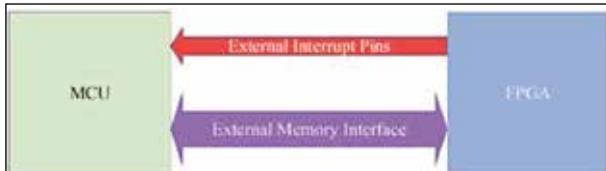
在本文結束之前，我們將證實彈性硬體架構會比更嚴謹的作法，更適合用來設計現代化的嵌入式系統。此外，此種作法還能改善專案成本及上市時間。論點、提供的範例以及個案研究都將用來辯護此立場。在此架構提供的設計彈性中，藉著觀察每個里程碑的價值，我們可清楚知道調適性硬體架構是推動嵌入式系統設計進步的強大助力。

## 探討輔助處理器架構的優點：設計彈性與高效能處理能力

FPGA 設計的常見應用之一是直接介接高速類比數位轉換器 (ADC)。訊號經過數位化處理，並讀取至 FPGA，然後會將一些數位訊號處理器 (DSP) 演算法套用到此訊號。最後，FPGA 會根據結果做出判斷。

此類應用將在整篇文章中當作範例。此外，圖 1 展示一般性輔助處理器架構，其中 MCU 和 FPGA 皆透過 MCU 的外部記憶體介面進行連接。FPGA 可當作外部靜態隨機存取記憶體 (SRAM) 的一部分。訊號從 FPGA 回到 MCU，並作為硬體中斷線路和狀態指示器使用。如此一來，FPGA 便能向 MCU

圖 1：一般性輔助處理器示意圖 (MCU + FPGA)。



圖片來源：CEPD

指出關鍵狀態。例如，通知 ADC 轉換已就緒，或是發生故障，或是發生另一個值得注意的事件。

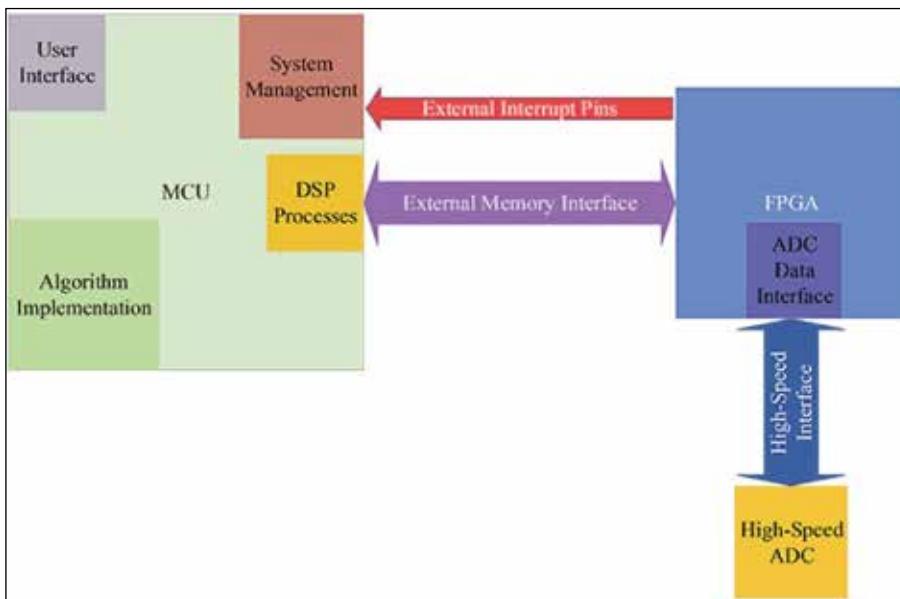
若想瞭解輔助處理器作法的優點，或許從前述各個里程碑的呈現項目最能看出。若要評估價值，除了列出工作或階段的成果外，還可評估這些成果所帶來的促進效果。下列問題的解答將有助於評估里程碑呈現項目的整體價值：

- 在專案的依存因素和瓶頸均已排除後，其他團隊成員的進度現在能加速進行嗎？
- 里程碑達到的成果如何進一步促成同步執行途徑？

### 使用微控制器進行數位訊號處理里程碑

此硬體架構允許的第一個開發階段就是將 MCU 擱在優先重要的位置。在一切條件均相同的情況下，若與 FPGA 和硬體描述語言 (HDL) 開發相比，

圖 2：架構 – 使用微控制器進行數位訊號處理。



圖片來源：CEPD

MCU 和可執行軟體開發所需的資源和時間會更少。因此，若以 MCU 作為主處理器開始產品開發，即可更快地實作、測試和驗證演算法。如此便能在設計流程早期發現演算法與邏輯程式錯誤，也能測試和驗證大部分的訊號鏈。

FPGA 在這個初期里程碑中的角色是高速資料收集介面。FPGA 的工作是從高速 ADC 中可靠地傳送資料、向 MCU 發出警報並通知有可用資料，並在 MCU 的外部記憶體介面上呈現此資料。雖然這個角色不包括實作 HDL 型 DSP 流程或其他演算法，但仍然極為關鍵。

在這個階段進行的 FPGA 開發作業，能為產品的開發工作以及上市時是否達到最終的成功，打下深厚的基礎。若只需關注在低層級的介面，便有足夠的時間可以專心投入於這些必要操作的測試。只要 FPGA 能可靠無虞地發揮介面角色，便定會達成這個里程碑。

這個初期里程碑的關鍵達成項目包括以下優點：

1. 完整訊號路徑，亦即所有的放大、衰減和轉換，都會經過測試和驗證。
2. 在初期實作軟體 (C/C++) 的演算法，便可減少專案開發的時間和工作量；這對管理階層和其他利

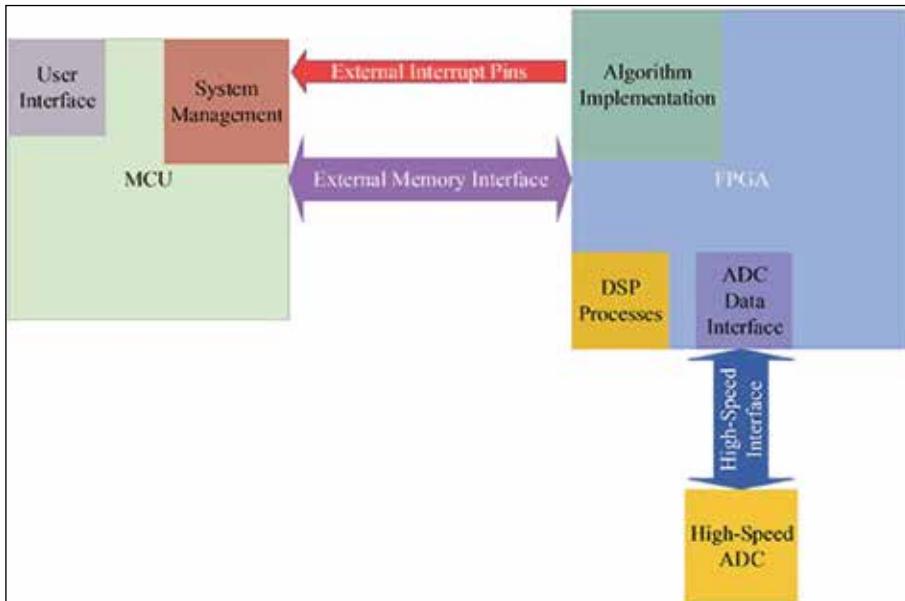
害關係方有相當大的價值，因為他們在核准未來設計階段前必須瞭解此專案是否可行。

3. 從 C/C++ 語言演算法實作中學到的經驗教訓，可直接轉移到 HDL 實作中 — 方法就是使用軟體對 HDL 工具，如 Xilinx HLS。

### 使用微控制器進行系統管理里程碑

這個輔助處理器作法提供的第二個開發階段，是將 DSP 流程和演算法實作從 MCU 移

圖 3：架構 – 使用微控制器進行系統管理。



圖片來源：CEPD

到 FPGA。FPGA 仍要負責處理高速 ADC 介面，不過若承擔其他角色，就可完全發揮 FPGA 提供的速度和同步執行能力。此外，與 MCU 不同的是，可實作並同時運行 DSP 流程和演算法通道的多個執行階段。

以 MCU 實作獲取的經驗教訓為基礎，設計人員即可滿懷信心邁向一個里程碑。以前述的 Xilinx Vivado HLS 為例，這些工具可提供從可執行 C/C++ 程式碼對可合成 HDL 的函式轉換。現在，計時限制、流程參數和其他使用者偏好，仍須定義和實作，不過核心功能已經保留並轉換到 FPGA 結構。

從這個里程碑開始，MCU 的角色便是系統管理員。FPGA 內的狀態和控制暫存器將由 MCU 監測、更新並回報。此外，MCU 還會管理使用者介面 (UI)。這個 UI 的形式，可以是透過乙太網路或 Wi-Fi 連線存取的網路伺服器，亦或是工業用觸控螢幕介面，讓使用者在使用點進行存取。MCU 這個全新更精細角色的關鍵優點是：MCU 和 FPGA 不必在處理運算密集型處理工作，而是用於兩者最擅長的工作中。

此里程碑的主要達成項目及其優點：

1. 由 FPGA 提供快速及同步執行的 DSP 流程和演

算法實作。MCU 提供簡化的回應式 UI，並可管理產品的流程。

2. 由於已先在 MCU 內完成開發與驗證，因此演算法的風險已經減輕，而且這些風險減輕措施也轉換到可合成 HDL 當中。藉助 Vivado HLS 之類的工具，這個轉換過程更加輕鬆。此外，還能透過整合式模擬工具 (如 Vivado 設計套件) 減輕 FPGA 特定的風險。

3. 利害關係方不會因為流程移到 FPGA 而暴露於重大風險中。相反地，他們可以看見並

享受 FPGA 的速度和同步作業所帶來的好處。現在可以觀察到可測量的效能改進，因此便可將注意力集中在全力準備好這個可製造性設計。

### 產品部署里程碑

由於運算密集型處理已 FPGA 內部進行，並由 MCU 負責系統管理和使用者介面職責，因此產品已作好部署的準備。現在，本文並非主張可略過 Alpha 和 Beta 測試版的發佈；不過，這個里程碑的著重點在於輔助處理器架構所提供的產品部署能力。

MCU 和 FPGA 都是現場可更新的元件。為了讓 FPGA 更新像軟體更新一樣容易，已經過數次改善。此外，由於 FPGA 是在 MCU 的可定址記憶體空間內，所以 MCU 可作為整個系統的存取點，亦即可接收針對其本身及 FPGA 的更新。可依據各個使用者，進行條件式的更新、排程、散佈和客製化。最後，使用者和使用案例記錄可加以維護，並能與特定版本實作相關聯。透過這些資料組，即使產品已部署在現場後，其效能仍可獲得持續的改進及加強。

或許這種全系統可更新的優點，如同空間型應

用中一樣，均不再獲得重視。一旦產品推出，就必須遠端進行維護和更新。這可能跟變更邏輯條件一樣簡單，或與更新通訊調變機制一樣複雜。FPGA 技術和輔助處理器架構所提供的可編程性，能涵蓋整個能力範圍，同時又可提供抗輻射元件選擇。

這個里程碑最終的主要好處是成本逐漸降低。成本降低、物料清單 (BOM) 變更及其他最佳化，都可在這個里程碑達成。在現場部署期間，或許會發現一個現象，產品若使用較便宜的 MCU 或能力較低的 FPGA，同樣也能地正常運作。由於輔助處理器的關係，架構設計人員不再受限於使用那些功能超出應用需求的元件。此外，萬一某個元件無法取得，此架構仍允許將新元件整合到設計中。但是，若使用單晶片、系統單晶片 (SoC) 架構，或使用會試著負責產品所有處理作業的高效能 DSP 或 MCU，情況就不是如此了。輔助處理器架構是功能和彈性的絕佳混合，能讓設計人員在開發階段和產品上市時，有更多的選擇與自由。

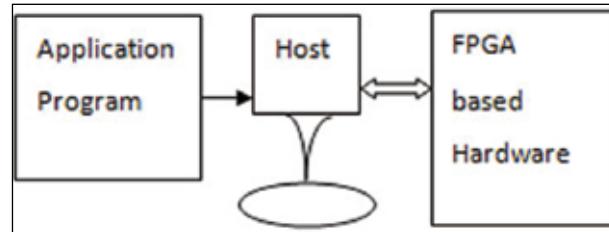
## 支援研究和相關的個案研究

### 衛星通訊範例

簡單地說，輔助處理器的價值在於減輕主處理單元的負荷，如此一來，就可在硬體上執行工作，並藉此享受加速和簡化的優勢。選擇這類設計的好處在於，運算速度和能力會淨增長，還有本文所主張的優點，減少開發成本和開發時間。可發揮這些優點的領域中，最受矚目的或許就是太空通訊系統。

G. Prasad 和 N. Vasantha 在 **FPGA based hardware as coprocessor** (FPGA 架構硬體做為輔助處理器) 發表文章中，詳細探討 FPGA 內部資料處理如何整合衛星通訊系統的運算需求，但不必花費對特殊應用積體電路 (ASIC) 進行一次性工程 (NRE) 的高額成本，而且對硬架構處理器也沒有應用專屬的限制。正如使用微控制器進行數位訊號處理里程碑中所述，他們的設計開始於應用處理器執行大部分的運算密集型演算法。從這個起始點，他們會判定軟體的哪些關鍵區域會消耗中央處理單

圖 4：衛星通訊範例中使用的應用程式、主機處理器和 FPGA 架構硬體。



元 (CPU) 的大多數時脈週期，並將這些部分轉移到 HDL 實作上。這種圖形呈現與目前現有的呈現具有高度相似性，但是他們選擇將應用程式呈現為自身的獨立區塊，因為可以在主機 (處理器) 或在 FPGA 架構硬體上實現。

利用周邊元件互連 (PCI) 介面，以及主機處理器的直接記憶體存取 (DMA)，周邊裝置的效能就會大幅提升。在去隨機性流程的改進中幾乎都可以觀察到此現象。當此流程在主機處理器的軟體中執行時，系統的即時反應顯然存在瓶頸。不過，轉移到 FPGA 後，即可觀察到以下優點：

- 去隨機化流程會即時執行，不會產生瓶頸
- 主機處理器的運算開銷已大幅降低，現在可以更好地發揮所需的記錄職責。
- 整個系統的總效能有所提升。

這一切工作都是在沒有 ASIC 相關成本的情況下完成，同時還可享有可編程邏輯的彈性 [5]。衛星通訊帶來相當大的挑戰，而上述方法可以滿足這些要求並獲得驗證，同時繼續提供設計彈性。

### 車用資訊娛樂範例

對挑剔的消費者而言，汽車內部的娛樂系統是區別性特點。與大多數汽車電子裝置不同的是，這些裝置十分顯眼，而且預期可提供出色的反應時間和效能。然而，設計人員常常在目前的設計需求以及未來功能所需的彈性之間左右為難。以這個例子來說，訊號處理和無線通訊的實作需求，將用來加強說明輔助處理器硬體架構的優勢。

採用的主流汽車娛樂系統架構之一，是由 Delphi Delco Electronics Systems 公司發佈。此架

構採用 SH-4 MCU 及配套的 ASIC，亦即 Hitachi 的 HD64404 Amanda 周邊裝置。這種架構能滿足汽車市場中 75% 以上的 basic 娛樂功能，但無法處理視訊處理應用和無線通訊。在這個現有架構中添加 FPGA，就可為這個現有設計方法增添更多的彈性和功能。

圖 5 所示的架構適用於視訊處理和無線通訊管理。將 DSP 功能推入 FPGA 後，Amanda 處理器

圖 5：車用資訊娛樂 FPGA 輔助處理器架構範例 1。

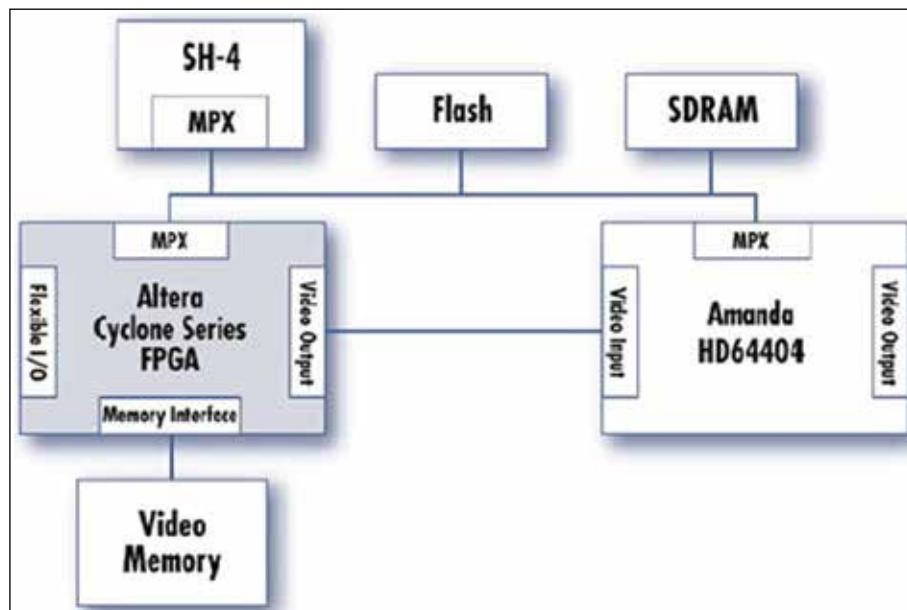
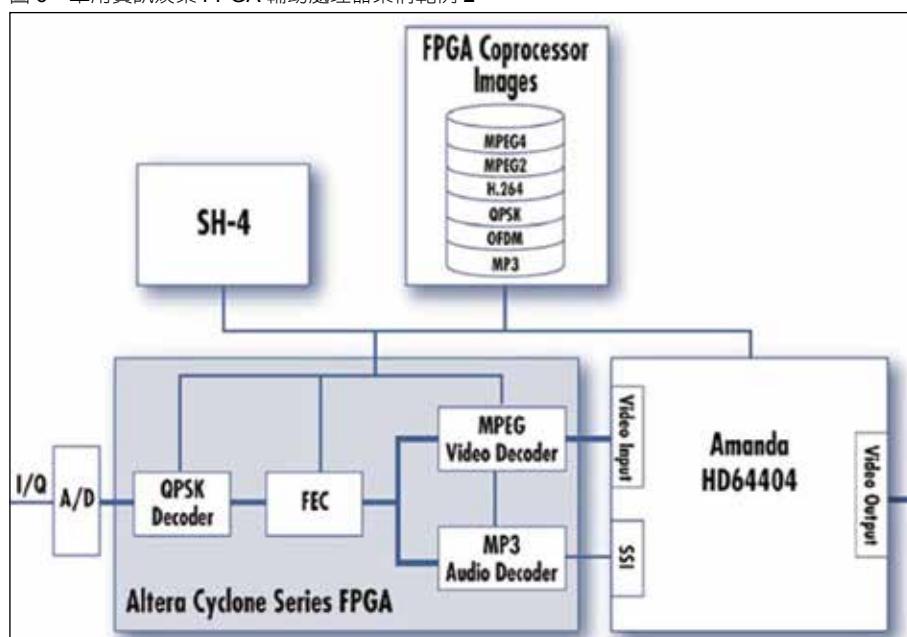


圖 6：車用資訊娛樂 FPGA 輔助處理器架構範例 2。



即可發揮系統管理的功用，並且可釋放出來以實作無線通訊堆疊。由於 Amanda 和 FPGA 都可以存取外部記憶體，因此可以在系統的處理器和元件之間快速交換資料。

圖 6 所示的第二個車用資訊娛樂重點在於，FPGA 有能力處理傳入的高速類比資料，以及視訊應用所需的壓縮和編碼工作。事實上，此功能全都能推入 FPGA 中，並且透過同步處理的運用，都可即時處理。

在現有硬體架構中納入 FPGA，既有硬體經過實證的效能，即可兼顧彈性並因應未來需求。即使在現有系統中，輔助處理器架構也可為設計人員提供多種選擇（原本是毫無其他選擇）[6]。

### 快速原型開發的優勢

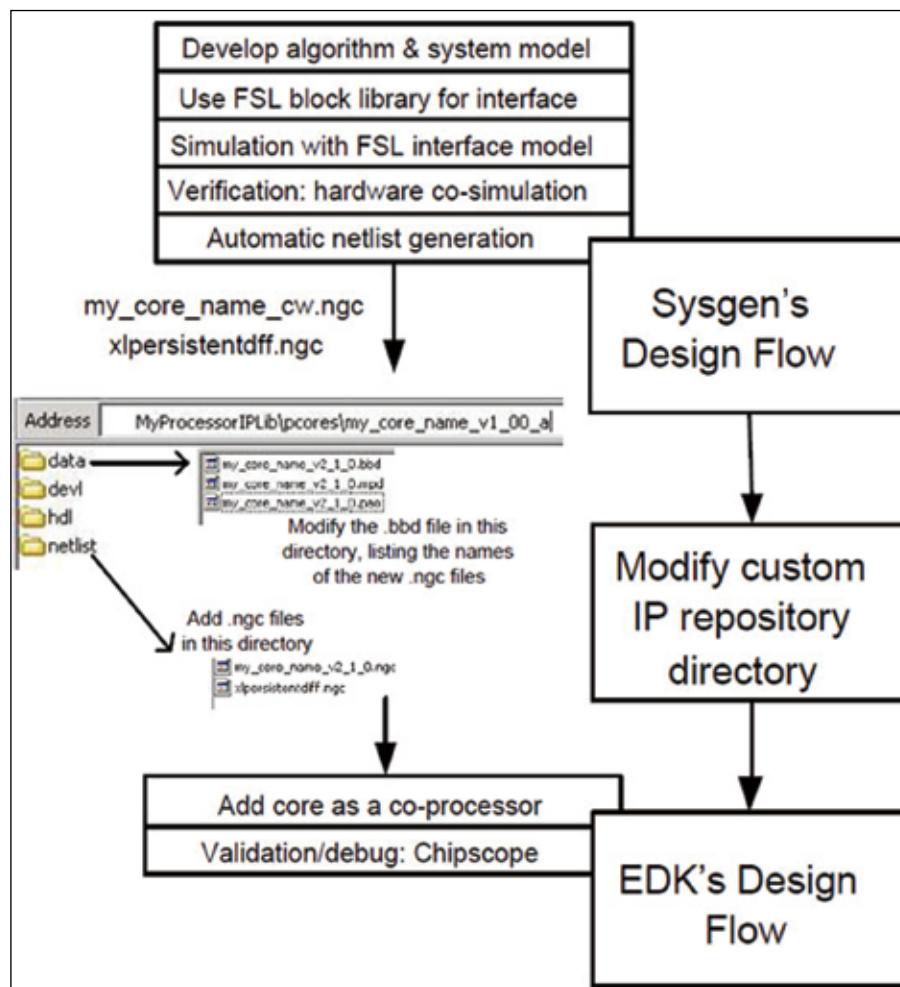
在本質上，快速原型開發流程透過同步執行多項工作、快速找出「程式錯誤」和設計問題，以及驗證資料和訊號路徑（特別是專案關鍵路徑中的路徑），來設法承擔產品開發領域中的大量工作。然而，為了讓這個流程真正產生精簡、高效的結果，那麼在必要的專案領域中必須有足夠的專業知識。

在過去，這表示必須有硬體工程師、嵌入式軟體或 DSP 工程師，以及 HDL 工程師參與其中。現在已有足夠的跨領域專業人員，他們或許能夠滿足多種職務的需求；但是，在協調這些工作方面，仍然需要大量的專案開銷。

在 An FPGA based rapid prototyping platform for wavelet coprocessors (適用於小波輔助處理器的 FPGA 型快速原型開發平台) 這篇文章中，作者們提出使用輔助處理器架構的想法，如此即可讓單一位 DSP 工程師以高效率且有效的方式滿足所有職務需求。針對這項研究，該團隊開始在 MATLAB 的 Simulink 工具中設計並模擬所需的 DSP 功能。這提供兩個主要功能：1) 透過模擬來驗證想要的效能；以及 2) 作為未來選擇設計時可供比較和參考的基準。

模擬之後，可找出關鍵功能並劃分成不同的核心，而這些核心是可在 FPGA 中合成的軟核心元件和處理器。這項工作中最重要的一步，是定義這些核心和元件之間的介面，並將資料交換效能與期望的模擬效能進行比較。此設計過程與 Xilinx 的嵌入

圖 7：實作設計流程。



式系統設計流程密切契合，其摘要如下圖 7 所示。

將系統劃分成可合成的核心，DSP 工程師即可專注在訊號處理鏈中最關鍵的部分。無需成為硬體或 HDL 方面的專家，即可修改、佈線或實作 FPGA 內不同的軟核心處理器或元件。只要設計人員瞭解介面和資料格式，就可以完全控制訊號路徑並改善系統效能。

## 實證發現—離散餘弦轉換的個案研究

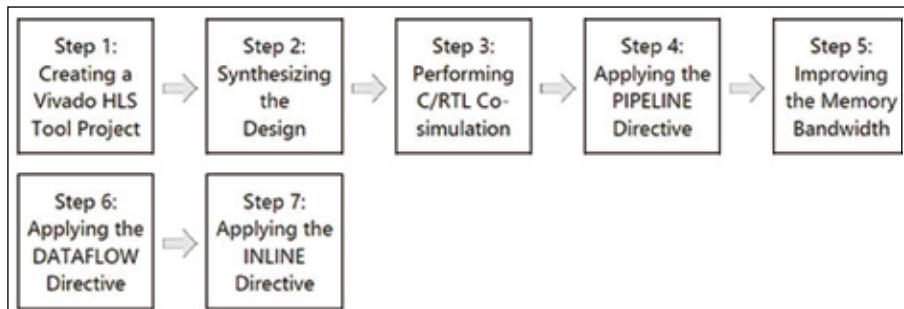
實證發現不僅證實輔助處理器架構可為嵌入式系統設計人員提供彈性，而且還展示現代 FPGA 工具可提供的效能增進選項。如下所述的增強功能可能無法使用，或對其他硬體架構的影響可能較小。離散餘弦轉換 (DCT) 被選為運算密集型演算法，而從 C 型實作進展到 HDL 型實作，則是這些發現的核心。由於此演算法在數位訊號處理中用於模式辨識和篩選，因此選擇 DCT [8]。這些實證發現來自於實驗室操作結果，是作者和其同事為了獲得 2020 - 2021 年 Xilinx 聯盟夥伴認證而發現。

在這項操作中使用了以下工具和元件：

- Vivado HLS v2019
- 評估和模擬所用的元件是 xczu7ev-ffvc1156-2-e

從 C 型實作開始，DCT 演算法接受兩個 16 位元數字陣列；陣列 a 是 DCT 的輸入陣列，陣列 b 是 DCT 的輸出陣列。因此，資料寬度 (DW) 定義為 16，陣列中的元素數 (N) 為  $1024/DW$ ，亦即 64。最後，DCT 矩陣 (DCT\_SIZE) 的大小設定為 8，意味著使用  $8 \times 8$  矩陣。

圖 8：Xilinx 的 Vivado HLS 設計流程。



C 型演算法實作遵循了本文的前提，能讓設計人員快速開發和驗證演算法的功能。儘管這是一個重要考量，但此驗證方式讓功能的權重高過執行時間的權重。這種權重方法是可允許的，因為此演算法的最終實作將在 **FPGA** 中進行，而硬體加速、迴路展開和其他技術都可立即取得。

只要在 Vivado HLS 工具中建立專案的 DCT 編碼後，下一步就是開始合成 **FPGA** 實作的設計。將演算法的執行從 **MCU** 轉移到 **FPGA** 中，在此下一個步驟中，有些最具影響力的優點會變得更加明顯。附帶說明，此步驟相當於上面所探討的使用微控制器進行系統管理里程碑。

現代 **FPGA** 工具能達到一套最佳化和增強功能，可大幅增進複雜演算法的效能。在分析結果之前，需要記住一些重要術語：

- **延遲** — 執行迴路所有迭代所需的時脈週期數 [10]
- **間隔** — 在下一次迴路迭代開始處理資料前的時脈週期數 [11]
- **BRAM** — 區塊隨機存取記憶體
- **DSP48E** — UltraScale 架構的數位訊號處理分割
- **FF** — 翻轉式
- **LUT** — 查找表
- **URAM** — 統一隨機存取記憶體 (可由單一電晶體組成)

## 預設值

預設的最佳化設定，取自於未經改

變的 C 型演算法轉換為可合成 HDL 後的結果。其中未啓用任何最佳化，因此可作為效能參考，以便進一步瞭解其他最佳化。

## 管線內部迴路

### Pipeline 指令會指示

Vivado HLS 展開內部迴路，以便在現有資料仍在管線內的時候，開始處理新資料。因此，新資料不必等到既有資料完成後才開始處理。

## 管線外部迴路

將 PIPELINE 指令套用到外部迴路，外部迴路的運算就算管線化。但是，內部迴路的運算現在會同時發生。將指令直接套用到外部迴路，延遲和間隔時間都會減少一半。

## 陣列分割

此指令會將迴路的內容映射到陣列，進而對這些陣列中單一元素的所有記憶體存取，全都進行扁平化。如此一來會消耗更多 RAM，但同樣地，這個

表 1：FPGA 演算法執行最佳化的發現結果 (延遲和間隔)。

*	延遲		間隔	
	最小值	最大值	最小值	最大值
預設 ( 解決方案 1)	2935	2935	2935	2935
管線內部迴路 ( 解決方案 2)	1723	1723	1723	1723
管線外部迴路 ( 解決方案 3)	843	843	843	843
陣列分割 ( 解決方案 4)	477	477	477	477
數據流 ( 解決方案 5)	476	476	343	343
直插 ( 解決方案 6)	463	463	98	98

表 2：FPGA 演算法執行最佳化的發現結果 (資源利用率)。

	BRAM_18K	DSP48E	FF	LUT	URAM
預設 ( 解決方案 1)	5	1	246	964	0
管線內部迴路 ( 解決方案 2)	5	1	223	1211	0
管線外部迴路 ( 解決方案 3)	5	8	516	1356	0
陣列分割 ( 解決方案 4)	3	8	862	1879	0
數據流 ( 解決方案 5)	3	8	868	1654	0
直插 ( 解決方案 6)	3	16	1086	1462	0

演算法的執行時間也會減少一半。

## 數據流

此指令能讓設計人員指定每次輸入讀數之間的目標時脈週期數。此指令僅支援頂層函數。只有向此層級開放的迴路和函數，才能從此指令中受益。

## 直插

**INLINE** 指令會將所有迴路扁平化，包括內部和外部迴路。行與列流程現在可以同時執行。所需的時脈週期數會保持在最小值，即便這會消耗更多 FPGA 資源。

## 結論

輔助處理器硬體架構為嵌入式設計人員提供高效能的平台，可在整個開發過程中以及產品發佈後，保持設計彈性。先驗證 C 或 C++ 語言的演算法，就可在相對較短的時間內驗證流程、資料和訊號路徑以及關鍵功能。然後，將處理器密集型演算法轉換成輔助處理器 FPGA，設計人員就可享受硬體加速和更加模組化設計的好處。

如果零件已停產或需要進行最佳化，相同的架構都可允許進行這些變更。新 MCU 和新 FPGA 可以納入設計中，同時，介面幾乎可以保持不變。此外，由於 MCU 和 FPGA 都可以現場更新，因此能在現場和遠端套用使用者特定的變更和最佳化功能。

最後，此架構結合 MCU 的開發速度和可用性，並兼具 FPGA 的效能和擴充性。由於在每個開發步驟中均可使用最佳化和效能增強功能，輔助處理器架構甚至可以滿足現今與未來設計中最具挑戰性的需求。

## 自動駕駛卡車新創公司透過 NVIDIA DRIVE 開發通用平台

自動駕駛卡車新創公司 Embark 正計畫開發具有高度適用性的人工智慧 (AI) 自動駕駛商用半掛卡車通用平台。

該公司宣布將透過 NVIDIA DRIVE 來開發各製造商皆能使用的 Embark 通用介面 (Embark Universal Interface : EU) 平台，當中包括自動駕駛卡車所需的運算和多模態感測器。這種極具彈性的做法，結合 NVIDIA DRIVE 的出色效能，創造出具有高度擴充性的解決方案，以提供更安全且高效的運輸物流服務。

大多數卡車運輸公司會使用多種車款，甚至會使用來自不同車廠的車輛，以在世界各地運輸各種貨物。Embark 平台可以整合至美國四大卡車製造商中的任何一個品牌，包含 PACCAR、Volvo、International 及 Freightliner。Embark 透過開發一個可以安裝到多款車輛的平台，幫助卡車業享受 AI 自動駕駛的優點，而無需等待特別為自動駕駛開發出的車輛。

從高速公路到擁擠的港口，自動駕駛車不斷吸收海量資料進行學習，以因應現實世界中難以預測的特性，而這種高速處理需要集中的高效能 AI 運算。

Embark 於四月時宣布與 Werner Enterprises、Mesilla Valley Transportation 和 Bison Transport 合作，以及包括 Anheuser Busch InBev 及惠普 (HP) 在內的托運客戶合作。

Embark 計畫在公開市場上市，於六月宣布成立特殊目的收購公司 (special purpose acquisition company : SPAC)，並攜手 Knight-Swift Transportation 共同合作。Embark 將加入 NVIDIA DRIVE 商業生態系成員的行列，這些成員透過公開上市共募集到超過 80 億美元。