從矽晶到軟體 - 設計低功耗嵌入式系統

第二部分 軟體設計原則

簡介

設計低功耗系統時，我們需要重視一些非傳統因素，這些因素涉及範圍從矽晶生產製程技術，到基於 MCU 的嵌入式平台上所執行的軟體。透過對系統層面的深入分析，本文討論決定 MCU 能源效率的三個關鍵參數：工作模式耗電、待機模式耗電和工作週期（Duty Cycle）– 決定各種狀態下所佔用的時間比率，由軟體本身來決定。

低功耗待機狀態使得 MCU 看起來能源效率極高，但事實是，只有考慮了控制工作模式功耗的所有因素後才能決定 MCU 的能源效率狀況。總之，對於處理製程、IC 架構和軟體結構選擇的權衡是十分微妙的決定因素，有時會出現意想不到的結果。此外，MCU 上功能模組結合的方式對整體耗電有著動態影響。即使對於硬體看似小而輕微的改變，都可能會導致系統執行週期中整體耗電的大幅波動。

本文的第一部分討論了在矽晶層面實現最低功耗的晶片級設計原則。

第二部分：軟體設計原則

效能擴充

如何實現節能高效的嵌入式應用，這有賴於軟體設計，而軟體設計又要以最合適的方式使用硬體資源，其中應用和硬體實現同樣重要。同樣的，處理器、時脈、電壓和記憶體利用等硬體的靈活性越大，開發人員可實現的節能潛力越大。此外，能感知硬體的軟體工具可幫助嵌入式系統工程師瞭解該如何逐步實現節能目標。

有一種方法是採用動態電壓調整，如圖 1 和圖 2 所示。此技術透過整合的 DC-DC 轉換器和效能監視電路來實現，當系統不需要以最高速率執行指令時，可降低供電電壓。如此一來，系統運行可處於低功耗狀態。好處在於其成為輸入電壓的函數，並且能在產品生命週期中改變。下圖顯示固定電壓（VDD 固定）、靜態電壓調整（SVS）和主動電壓調整（AVS）之間的相對差異。

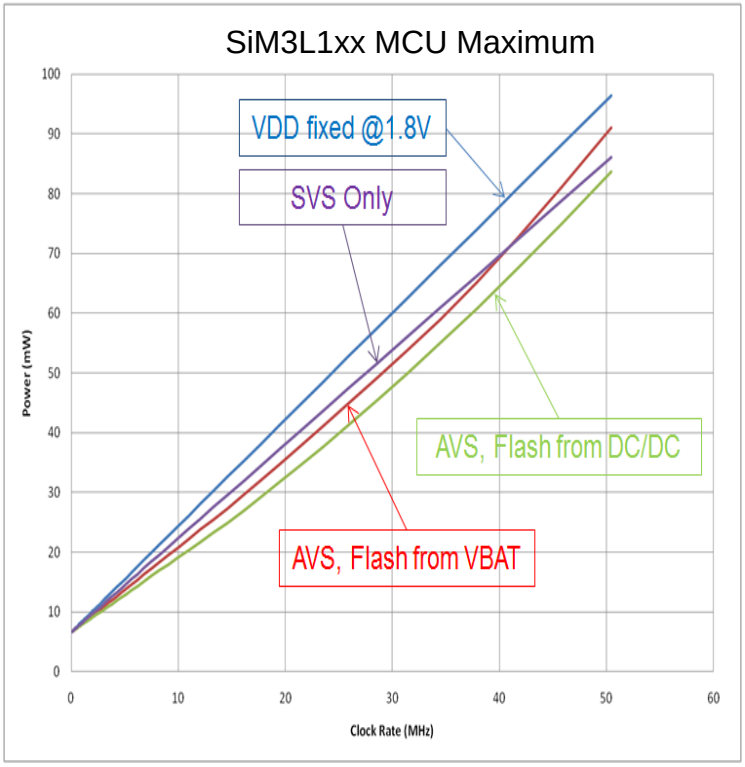

圖 1：VBAT=3.6V 時電壓調整效率

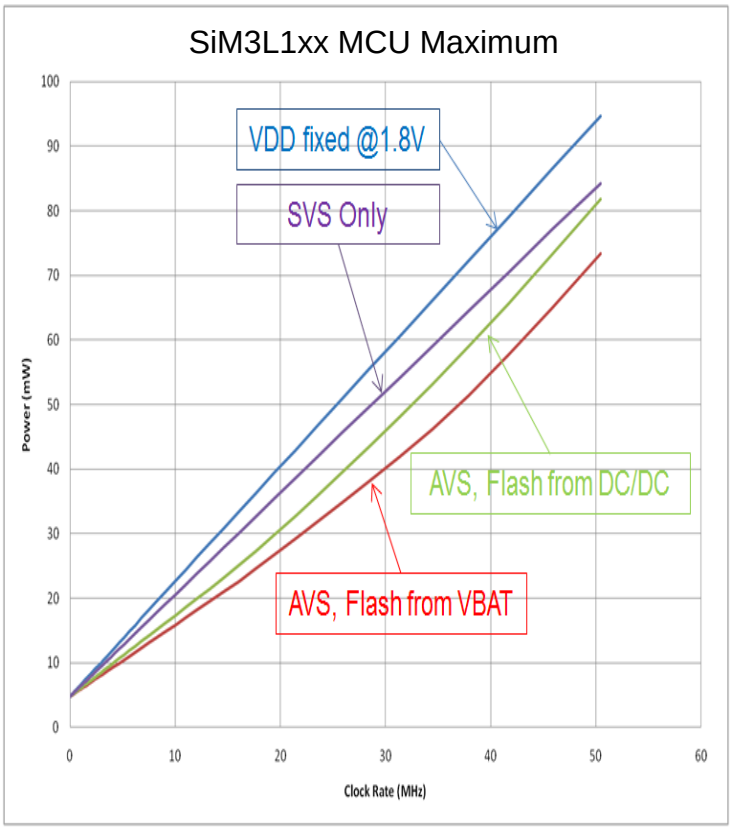

圖 2：VBAT=2.4V 時電壓調整效率

AVS 有一個令人感興趣的現象，即 AVS 能夠隨著輸入到系統中的電壓而改變。在這個範例中，當輸入電壓為 3.6V 時，採用內部 DC-DC 轉換器為內部邏輯（也包括 Flash 記憶體）供電效率更高。然而，當輸入電壓降低以後（例如電池在產品生命週期中放電），採用輸入電壓直接為 Flash 記憶體子系統供電是更高效率的方法，因為此時與記憶體相比，內部邏輯能夠在更低的電壓下運行。例如來自 Silicon Labs 的新型 SiM3L1xx 低功耗 32 位元 MCU 系列產品具有靈活的供電架構，支援六個獨立和可變的供電電源（power domain），可實現這種動態最佳化。

大部分的情況下，當電壓降低時 CMOS 邏輯電路操作變慢。如果應用可容忍更低效能（一般處理通訊協定時，所需的資料速率不超過特定標準化頻率），那麼低電壓所帶來的耗電降低則更加省電（以平方倍數降低）。漏電損耗提供電壓調整的下限。如果操作花費太多時間，那麼漏電損耗將開始主導電源消耗方程式，並且增加整體功耗。由於這個原因，為了使漏電元件功耗最小化，應該儘快執行操作，然後使處理器進入休眠模式。

假設一個無線感測器應用，其需要進行大量的數位訊號處理（DSP），以玻璃破碎檢測器為例。在這個範例中，使用快速傅立葉變換（FFT）分析由音訊感測器接收到的變動（特性頻率由玻璃破碎產生）。FFT 是一個相對複雜的演算法，因此透過降低電壓而得到的低頻下運行該演算法，可能顯著增加漏電損耗，即使採用舊的生產製程技術也一樣。在這種情況下，最好的方法是在接近最大頻率下運行，然後迅速返回到休眠狀態，一直到需要向主機報告時再運行。

然而，無線協定代碼要滿足不同的需求。無線協定有固定的事件時序。在這些情況下，協定可完全由硬體處理，進而降低處理器核心電壓。因此，封包組合和傳輸代碼可運行在適合於無線協定的速率。

附加的硬體模組，例如智慧型直接記憶體存取（DMA），能夠進一步影響電源消耗。許多 DMA 控制器，例如由 ARM® Cortex™-M3 處理器提供的 DMA 控制器，與處理器之間互動頻繁。然而，更多的智慧型 DMA 控制器同時支援順序和連結，允許處理器計算封包標頭（packet header）、加密資料、封包組合，然後以適當的間隔時間把資料封包發佈到由射頻前端使用的記憶體緩衝區中。因為射頻鏈路的大多數時間是處於活動狀態，處理器能夠進入休眠模式，進一步節省大量的能量。

記憶體的利用

採用現代 32 位元 MCU 元件，軟體工程師能比較自由的使用記憶體區塊。通常，MCU 提供混合儲存方案，非揮發性 Flash 記憶體用於長期代碼儲存，而靜態隨機存取記憶體（SRAM）用於保留臨時資料。大多數情況下，存取 Flash 儲存區所產生的功耗大於存取 SRAM 儲存區所產生的功耗。在正常使用過程中，Flash 儲存區讀取時間是 SRAM 讀取時間的三倍。Flash 寫入操作需要先把整塊區域清除，然後使用較長的相對高壓脈衝序列重新定義，消耗很多能量；然而對於大多數應用來說，Flash 寫入操作很少發生，不會顯著影響平均耗電。

影響 Flash 記憶體功耗的另一個因素是如何分配處理器存取方式。Flash 記憶體的每個區塊由多個頁面（page）組成，通常每頁最大為 4kB。為了支援任意存取，所有頁面都要上電；任何未使用的頁面都維持在低功耗狀態。

如果經常存取的程式碼橫跨兩個 Flash 頁面而非在一個頁面中，那麼指令讀取相關的功耗將增加。為頻繁存取的程式和資料重新分配非關聯的頁面，能夠節省大量電池耗電，且不需要對實體層硬體進行改變。

通常明智的做法是把頻繁使用的功能代碼複製到 SRAM，從 SRAM 而非從 Flash 中讀取這些代碼指令，雖然這樣看起來記憶體效率有所降低，然而電池壽命的延長可彌補略高的記憶體消耗。

程式最佳化

能源最佳化也顛覆了傳統觀念上的程式效率。幾十年來，嵌入式系統工程師都集中在為記憶體大小而最佳化程式代碼。能源最佳化提供了一套全新指標，一個重要的考慮因素是晶片內部緩衝區的利用，這通常存在於 32 位元 MCU 平台。

程式代碼大小的最佳化使緩衝區中可保留更多可執行代碼，進一步改善速度和功耗。然而，重複使用的公共代碼，包括過去常被用來減少應用代碼大小的 function call 和 branch，會在高速緩衝區中相同行上導致意想不到的衝突。當指令從主儲存區中提取多個 Flash 頁面啟動時，將導致高速緩衝區衝突以及多 Flash 頁面啟動。

對於在產品整個生命週期中經常運行的代碼，比較妥善的作法是把除了 branch 或 function call 以外的代碼壓縮到緩衝區。例如煙霧警報器。即使每週觸發一次警報（可能由廚房中的過量煙霧引起），在警報器 10 年壽命 3.15 億次檢查中僅有 520 次事件發生。大部分時間中，代碼僅僅執行感測器讀取，檢查是否超出門限，然後處理器核心處於休眠狀態直到被系統計時器再次喚醒。

在警報系統執行的所有感測器讀取之外，不到 0.0002%的機率會導致警報產生代碼執行。剩下 99.9998%的時間處於核心感測器讀取循環。使功耗最小化的關鍵在於確保警報產生代碼不在緩衝區執行序列中運行。因為他較少運行，可採用更傳統的技術優化這類代碼。

高效節能工具

為了使 MCU 平台的能源效率發揮到極致，工具支援至關重要。把功能分配到 Flash 上非關聯頁面中，這需要連接器清楚每個目標 MCU 的詳細記憶體映射。連接器能夠根據開發

者的輸入來判別模組是否可跨越頁面邊界，並為非揮發性儲存產生最高效節能的二進位碼。原則上，這個代碼也用於確保函數和資料被這樣放置 - 最常用的執行函數不會在緩衝區中產生衝突。由於 MCU 供應商所提供的工具清楚知道每個目標平台的儲存區版面配置和功耗需求，以上細節層次更容易實現，而對於協力廠商來說則相當困難。

MCU 非常清楚如何構成不同的周邊和匯流排。這種知識能夠應用在工具中指導工程師進行選擇，且不會浪費電源。由 Silicon Labs 開發的圖形化 AppBuilder 環境就是一個典型範例，如圖 3 所示。此工具可透過拖拽周邊來為應用定義軟體框架。

AppBuilder 能夠檢視周邊裝置設置並決定是否可進行節能調整。例如，如果使用者把 UART 增加到應用，並將速率設定為 9600 baud，則工具將檢查 UART 周邊裝置匯流排並進行適當的設定。ARM 周邊裝置匯流排（APB）用於主機模組（例如 UART 和類比數位轉換器）能夠運行在最大 50MHz 速率下。在這個範例中，該速率遠遠高於（並會消耗更多的能量）所需的速率；因此工具要求使用者是否想減少 APB 的資料速率到一個更加適合的等級。

此外，AppBuilder 軟體為工程師提供與功效相關的特殊應用資訊。使用目標 MCU 的模擬（也能夠透過瞭解詳細的矽晶特性來實現），工具可提供互動式圖表以顯示電流消耗，不僅僅對於整個應用，同時也用於處理器和每個周邊裝置。

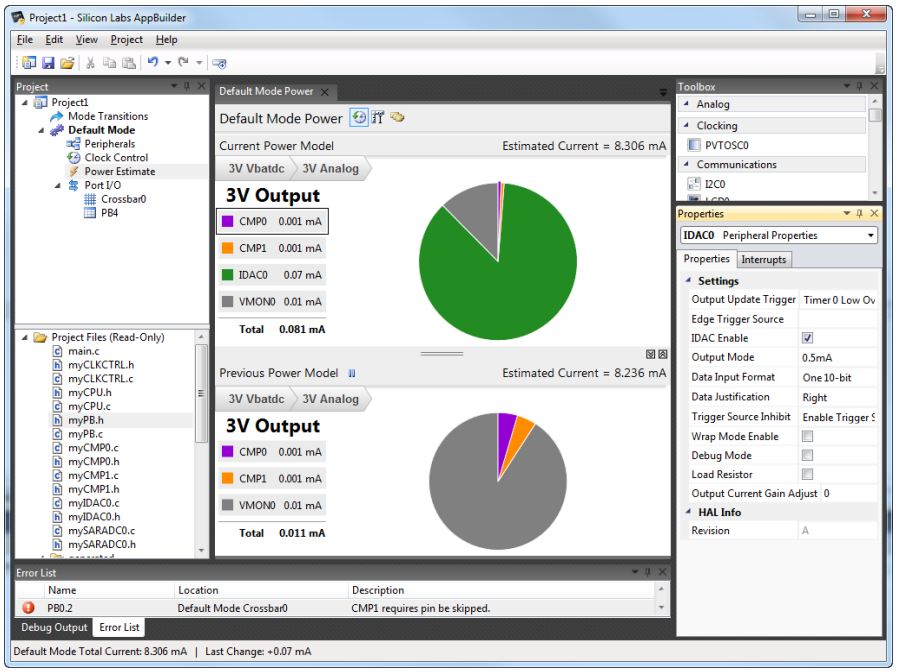

圖 3：具有耗電感知工具的 AppBuilder 介面截圖

開發工具將會演變並具有更多「功耗感知」能力。傳統上，偵錯特性（例如中斷點）被設定為事件（例如記憶體讀寫事件）。未來，中斷點支援會用作處理功耗相關的問題。例如，如果在特殊界限或自上次休眠狀態後所產生的累積功耗超過目標值，偵錯器將觸發並顯示出在應用中哪些部分耗電超過預期（例如，跨越 Flash 頁面邊界的代碼可能更頻繁的超過預計）。比預計更高的功耗以及在記憶體映射中代碼位置上的資訊，可提供關鍵線索，進一步幫忙軟體開發人員採用適當措施。

結論

低功耗系統設計是一個整體設計過程，需要選擇合適的矽晶、軟體和開發工具。透過掌握這些變數因素之間的關係，系統工程師可開發出更為高效節能的嵌入式系統，突破電池供電型應用的諸多限制。

# # #

Silicon Labs致力於投資研究與開發，以幫助我們的客戶採用創新的低功耗、小尺寸、類比密集型混合訊號解決方案開發差異化的市場產品。Silicon Labs廣泛的專利組合證明我們具有獨特的發展方式和世界一流工程團隊。專利查詢：www.silabs.com/patent-notice。

© 2012 Silicon Laboratories Inc.、ClockBuilder、DSPLL、Ember、EZMac、EZRadio、EZRadioPRO、EZLink、ISOmodem、Precision32、ProSLIC、QuickSense、Silicon Laboratories和Silicon Labs 標示是Silicon Laboratories Inc.的商標或注冊商標。ARM和Cortex-M3是ARM 控股公司的商標或注冊商標。ZigBee是ZigBee Alliance, Inc.的注冊商標。所有其他產品名稱可能各自屬於對應公司的商標。