



DSP builder 是干啥的？

DSPbuilder 就是 Altera 設計用來方便客戶設計 DSP 的工具。首先我們了解一個背景，現下最牛的算法軟體毋庸置疑的就是 Matlab 了。或許在統計學，或者其他的數學模型方面有更加專業的工具，但是對於 DSP 而言，Matlab 是不二的選擇。N 多的人在用他來搭建模型。我們可以把 Matlab 分成 matlab 和 simulink 兩部分。Matlab 更多的是對數組進行一系列的計算，而這種計算是靜態的。而 simulink 是使用模塊化的模式來搭建一個模型，而這個模型才是動態化的。當我們用 matlab 做成一個算法的時候，先不用開心太早，因為它完全有可能是無法實現的。更多的只是一種理論上的公式。只有當我們用 simulink 來搭建出一個模型的時候，我們可以知道，它是可以被實現出來的，無論是軟體，或者硬體的模式。

我們小的時候學數學，學乘法，大家一定還有美好的回憶，個位乘，然後進位到十位，然後在一步一步的算下去，算了長長的一溜，很有成就感的。我記得那個時候我們還要用尺來畫中間的那根線呢，因為老師說那條線要畫得直，好像畫不直的話就算不出正確答案了（大概所謂心誠則靈的意思吧）。後來知道了，這個世界上，居然有一個東西叫做電算機。那個東西是你無論如何光著腳都攆不上的速度，而且不會出錯（不過好像也見過 $1+1=3$ 的電算機）。我們於是發現自己似乎被騙了，我們為什麼要做那一長串的中間過程呢？是的。如果我們用這種模式來考慮問題的話，一邊是做好的算法模型，一邊是我們的 FPGA 晶片。為什麼我們還需要很辛苦的寫什麼 Verilog, VHDL 代碼呢？就好像在學 Verilog, VHDL 之前我們做什麼？我們學數字電路，現下還有幾個人在用那些與非門搭電路呢？所謂社會的進步一定是不斷從低端向高端發展的歷史過程。所以，與非門對於 Verilog 來說是沒必要的中間過程。但其實 Verilog 本身又何嘗不是呢？當然了，Verilog, 與非門這些東西，對於我們對電路的理解和學習是做出過貢獻的，但是，是不是到了應該把它們淘汰了的時候了呢（我這裡說的只是和算法相關的電路）？或許現下還沒有真的到了這個時期，但是它正在一步步的向我們靠近。而 DSP builder 就是基於這種思想的工具。讓我們看看用了 DSP builder, 可以給我們省多少東西。

首先是省了時間，時間是這個功利社會最重要的東西，沒有人有耐心等待你做一個十全十美的東西出來。誰最早出來，哪怕是東西爛到不能用，都是成功的。所以時間比什麼都重要。然後是省了人，不需要找一些人來把那些算法轉變成代碼了。如果大家對 Simulink 比較熟悉的話，可以看到，simulink 的庫裡面有你的一切實驗環境。好比信號發生器啊，信道啊。另外還有最全面的檢測系統，比如說示波器啊，頻譜分析儀啊，誤碼率分析器啊。有了這一切，你還要實驗室做什麼？至少你不需要那一堆昂貴的儀器了吧。

物盡其用，人盡其能，我們還是希望盡量多的去發揮一些優勢性的東西。所以千萬不要說用 DSP builder 去做一個非常複雜的控制邏輯，那只能是得不償失。所以，我更建議大家用 DSP builder 來做一些算法方面的實現。然後作為整個系統中的一個模塊來使用。這樣可以使你的設計更有效率，並且可修改性增強。DSP builder 和 SOPC builder 的結合使用可以大大增強你的設計能力。比如說，我們可以用 DSP builder 來做一些運算模塊，然後再 SOPC builder 裡面連接起來。再用一個 NiosII 來作為控制，寫一些軟體代碼。一個看上去非常複雜的系統，其實已經可以看到了。而這整個過程中，我們沒有寫過一行硬體代碼。可以說，這種系統設

計思路就是未來的一個趨勢。

DSP builder 設計看上去簡單，但要精通也並不容易。有兩個方面是比較讓人困惑不已的。如果你是一個做算法的人，你對 Simulink 非常熟悉，而你對硬體方面所知寥寥，那有可能犯第一個問題：時序問題。在設計中，你很難找到時序的節點，就是暫存器，因為暫存器都是存在於模塊本身裡面的。有很多人設計了一個從頭到尾都是組合電路的設計。而這種設計是肯定無法透過時序驗證的。解決這樣的問題，現下有兩個模塊可以幫助。第一個是 8.0 裡面新出來的，Display Pipeline Depth。很明顯，這個模塊的作用就是現實所有模塊的流水級數，就是這個模塊裡面有幾層暫存器。這樣你從模型上就可以很清楚的看到兩級暫存器之間的距離，規避冗長的組合電路。另外一個就是 Resource Usage。這個是資源使用顯示模塊，在編譯以後，可以顯示資源的使用量。但是它有一個附屬功能，其實更加有用，就是 timing。它可以用來顯示所有沒有透過時序分析的路徑，然後在模型中間 highlight 出來。這樣可以幫助你很容易的發現問題所在，然後做些亡羊補牢的事情。

針對對硬體非常熟悉，但是對 Simulink 這種東西非常不熟悉的人有一個麻煩，那就是時鐘。在 DSPbuilder 的設計中，所有的時鐘和時鐘復位信號都是隱藏的。在 simulink 中，時鐘是透過採樣率來定義的，或者叫時鐘域。從一個初始的採樣率開始，後面的模塊都是使用相同的時鐘域的。我們可以使用 Rate Change 目錄下的模塊來對時鐘進行切換。而這似乎讓人很容易暈。這裡我們可以使用 simulink 的工具。它可以設定使得不同時鐘域的模塊顯示成爲不同顏色。這樣看起來就舒服很多了。

再說一個 DSP builder 裡面最讓人心動的東西吧。HIL (hardware in loop)。這是一個非常有趣的觀念。我們現下討論的是一個 FPGA 平台，這個晶片是可以無限次燒錄的。不會燒一次收你一塊錢的。而我們在做設計的時候經常有這樣的問題，對設計本身沒有把握，不知道是不是對，當然可以做仿真。但是仿真有的時候很麻煩，一個是速度可能會太慢了點。一個是我依然不相信你們的這個仿真模型是不是真的夠仿真。所以，你就可以嘗試用 HIL 來加速你的仿真，同時增強你的信心。HIL 把你的設計包裹在一套界面中間，進行編譯，然後下載到板子的 FPGA 中間。Simulink 透過下載電纜把測試數據不斷的灌入，然後在輸出端不斷的獲得硬體跑出來的結果。透過這麼一個過程，保證你的仿真達到多快好省的境界。但是，這也只是仿真而已，我不管它叫做測試，因為它不是在真實的時鐘頻率下操作的。時鐘是透過 Jtag 的時鐘驅動的。所以你不要指望它可以跑在 100,200 兆上。但是速度依然已經夠快了。如果這樣的電路下載下去還有問題，那可能性只有一個，你的約束錯了。或者是時序有問題，或者是你的管腳出問題。而你的算法設計本身不會有問題。