



DSP builder 是干啥的？

DSPbuilder 就是 Altera 设计用来方便客户设计 DSP 的工具。首先我们了解一个背景，现在最牛的算法软件毋庸置疑的就是 Matlab 了。或许在统计学，或者其他的数学模型方面有更加专业的工具，但是对于 DSP 而言，Matlab 是不二的选择。N 多的人在用他来搭建模型。我们可以把 Matlab 分成 matlab 和 simulink 两部分。Matlab 更多的是对数组进行一系列的运算，而这种计算是静态的。而 simulink 是使用模块化的方式来搭建一个模型，而这个模型才是动态化的。当我们用 matlab 做成一个算法的时候，先不用开心太早，因为它完全有可能是无法实现的。更多的只是一种理论上的公式。只有当我们用 simulink 来搭建出一个模型的时候，我们可以知道，它是可以被实现出来的，无论是软件，或者硬件的方式。

我们小的时候学数学，学乘法，大家一定还有美好的回忆，个位乘，然后进位到十位，然后在一步一步的算下去，算了长长的一溜，很有成就感的。我记得那个时候我们还要用尺来画中间的那根线呢，因为老师说那条线要画得直，好像画不直的话就算不出正确答案了（大概所谓心诚则灵的意思吧）。后来知道了，这个世界上，居然有一个东西叫做计算器。那个东西是你无论如何光着脚都撵不上速度，而且不会出错（不过好像也见过 $1+1=3$ 的计算器）。我们于是发现自己似乎被骗了，我们为什么要做那一长串的中间过程呢？是的。如果我们用这种方式来考虑问题的话，一边是做好的算法模型，一边是我们的 FPGA 芯片。为什么我们还需要很辛苦的写什么 Verilog, VHDL 代码呢？就好像在学 Verilog, VHDL 之前我们做什么？我们学数字电路，现在还有几个人在用那些与非门搭电路呢？所谓社会的进步一定是不断从低端向高端发展的历史过程。所以，与非们对于 Verilog 来说是没必要的中间过程。但其实 Verilog 本身又何尝不是呢？当然了，Verilog, 与非门这些东西，对于我们对电路的理解和学习是做出过贡献的，但是，是不是到了应该把它们淘汰了的时候了呢（我这里说的只是和算法相关的电路）？或许现在还没有真的到了这个时期，但是它正在一步步的向我们靠近。而 DSP builder 就是基于这种思想的工具。让我们看看用了 DSP builder, 可以给我们省多少东西。

首先是省了时间，时间是这个功利社会最重要的东西，没有人有耐心等待你做一个十全十美的东西出来。谁最早出来，哪怕是东西烂到不能用，都是成功的。所以时间比什么都重要。然后是省了人，不需要找一些人来把那些算法转变成为代码了。如果大家对 Simulink 比较熟悉的话，可以看到，simulink 的库里面有你的一切实验环境。好比信号发生器啊，信道啊。另外还有最全面的检测系统，比如说示波器啊，频谱分析仪啊，误码率分析器啊。有了这一切，你还要实验室做什么？至少你不需要那一堆昂贵的仪器了吧。

物尽其用，人尽其能，我们还是希望尽量多的去发挥一些优势性的东西。所以千万不要说用 DSP builder 去做一个非常复杂的控制逻辑，那只能是得不偿失。所以，我更建议大家用 DSP builder 来做一些算法方面的实现。然后作为整个系统中的一个模块来使用。这样可以使你的设计更有效率，并且可修改性增强。DSP builder 和 SOPC builder 的结合使用可以大大增强你的设计能力。比如说，我们可以用 DSP builder 来做一些运算模块，然后再 SOPC builder 里面连接起来。再用一个 NiosII 来作为控制，写一些软件代码。一个看上去非常复杂的系

统，其实已经可以看到了。而这整个过程中，我们没有写过一行硬件代码。可以说，这种系统设计思路就是未来的一个趋势。

DSP builder 设计看上去简单，但要精通也并不容易。有两个方面是比较让人困惑不已的。如果你是一个做算法的人，你对 Simulink 非常熟悉，而你对硬件方面所知寥寥，那有可能犯第一个问题：时序问题。在设计中，你很难找到时序的节点，就是寄存器，因为寄存器都是存在于模块本身里面的。有很多人设计了一个从头到尾都是组合电路的设计。而这种设计是肯定无法通过时序验证的。解决这样的问题，现在有两个模块可以帮助。第一个是 8.0 里面新出来的，Display Pipeline Depth。很明显，这个模块的作用就是现实所有模块的流水级数，就是这个模块里面有几层寄存器。这样你从模型上就可以很清楚的看到两级寄存器之间的距离，规避冗长的组合电路。另外一个就是 Resource Usage。这个是资源使用显示模块，在编译以后，可以显示资源的使用量。但是它有一个附属功能，其实更加有用，就是 timing。它可以用来显示所有没有通过时序分析的路径，然后在模型中间 highlight 出来。这样可以帮助你很容易的发现问题所在，然后做些亡羊补牢的事情。

针对对硬件非常熟悉，但是对 Simulink 这种东西非常不熟悉的人有一个麻烦，那就是时钟。在 DSPbuilder 的设计中，所有的时钟和时钟复位信号都是隐藏的。在 simulink 中，时钟是通过采样率来定义的，或者叫时钟域。从一个初始的采样率开始，后面的模块都是使用相同的时钟域的。我们可以使用 Rate Change 目录下的模块来对时钟进行切换。而这似乎让人很容易晕。这里我们可以使用 simulink 的工具。它可以设置使得不同时钟域的模块显示成为不同颜色。这样看起来就舒服很多了。

再说一个 DSP builder 里面最让人心动的东西吧。HIL (hardware in loop)。这是一个非常有趣的概念。我们现在讨论的是一个 FPGA 平台，这个芯片是可以无限次烧录的。不会烧一次收你一块钱的。而我们在做设计的时候经常有这样的问题，对设计本身没有把握，不知道是不是对，当然可以做仿真。但是仿真有的时候很麻烦，一个是速度可能会太慢了。一个是我依然不相信你们的这个仿真模型是不是真的够仿真。所以，你就可以尝试用 HIL 来加速你的仿真，同时增强你的信心。HIL 把你的设计包裹在一套接口中间，进行编译，然后下载到板子的 FPGA 中间。Simulink 通过下载电缆把测试数据不断的灌入，然后在输出端不断的获得硬件跑出来的结果。通过这么一个过程，保证你的仿真达到多快好省的境界。但是，这也只是仿真而已，我不管它叫做测试，因为它不是在真实的时钟频率下操作的。时钟是通过 Jtag 的时钟驱动的。所以你不要指望它可以跑在 100,200 兆上。但是速度依然已经够快了。如果这样的电路下载下去还有问题，那可能性只有一个，你的约束错了。或者是时序有问题，或者是你的管脚出问题。而你的算法设计本身不会有问题。