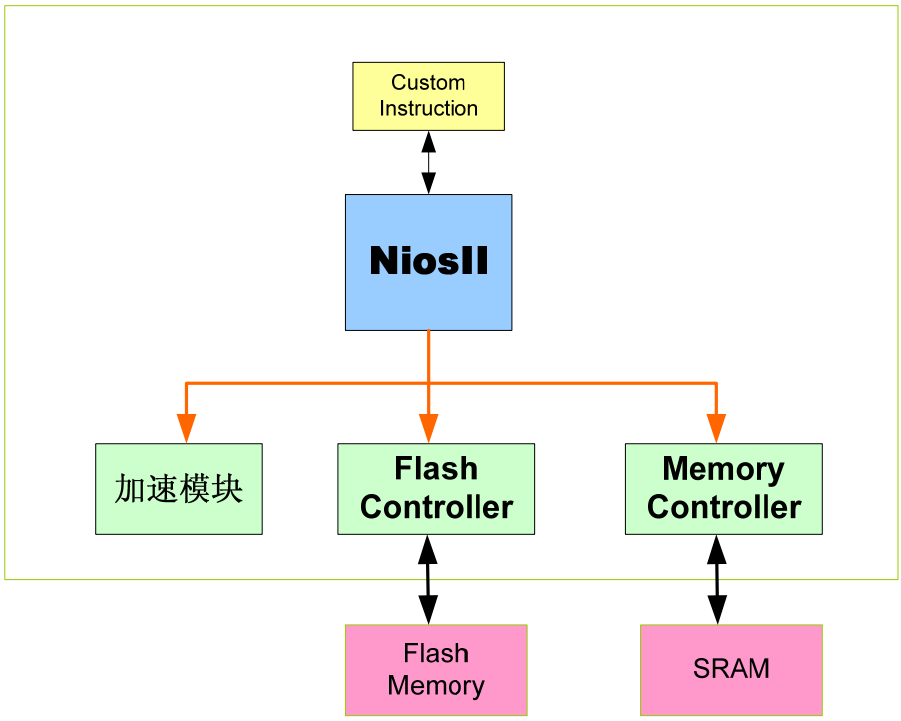
系统的问题，零零碎碎说了那么些，最终的目的，其实就是要把积木搭起来，而我们之前说的都是一些工具，我们组建一个系统可以使用的一些资源。看看我们是否可以根据我们对这些积木的了解来组建一个最炫的系统出来。我们首先看看我们有些什么好了。

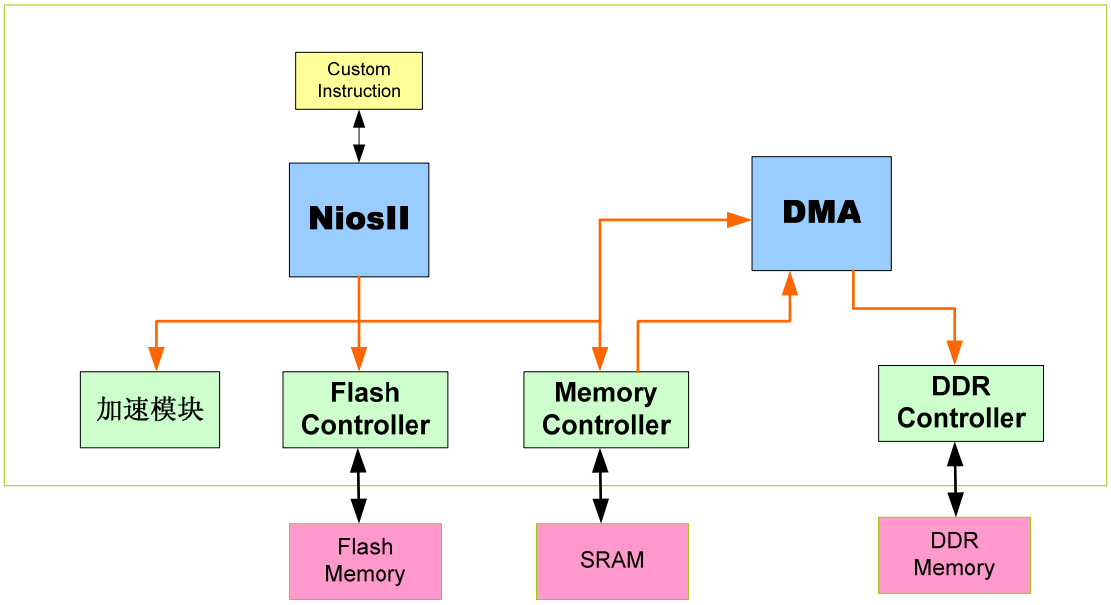
- ***NiosII：牛死 or 死牛***

NiosII 其实是个很好玩的东西，他可以让你的系统牛死，但是也很容易就变成了死牛。NiosII 作为一个软核，好处是可以很容易的实现多核系统，坏处就是他的速度不够快，一百多兆的速度，显然和什么 ARM 啦什么的没办法比了。但是问题是，NiosII 是在 FPGA 中实现的，所以他可以使用非常多的加速方式，使得这头死牛变得牛死。作为 CPU，它本身的控制，灵活性是毋庸置疑的，所以我们需要在操作速度上面做文章。把一个几十，甚至几百行的软件代码，变成硬件，成为一条指令。对了，我们可以使用自定义指令来实现。比如一个比较庞大的复杂运算，而输入和输出又相对单纯。这当然是一种方式，这种方式相对来说是最简单的，因为软件里面只是调用了一条指令而已。但是这种方式有一个小缺点，当你的运算非常大的时候，CPU 不得不停下来等着你完成计算它才会继续下去。针对这种状况，我们可选择使用一个 component 来加速。就是把加速硬件模块做成一个 component，然后用 Avalon-MM 和它进行连接。这样在这个模块进行运算的时候，CPU 并不会暂停，而是不断的继续下面的程序。最后 CPU 来看一下结果就好了。但是坏处是，会占据一段时间的 CPU 上的 avalon-mm 端口。硬件加速的效果是可以惊人的，完全可以做到让一条死牛，变成真正的牛死。



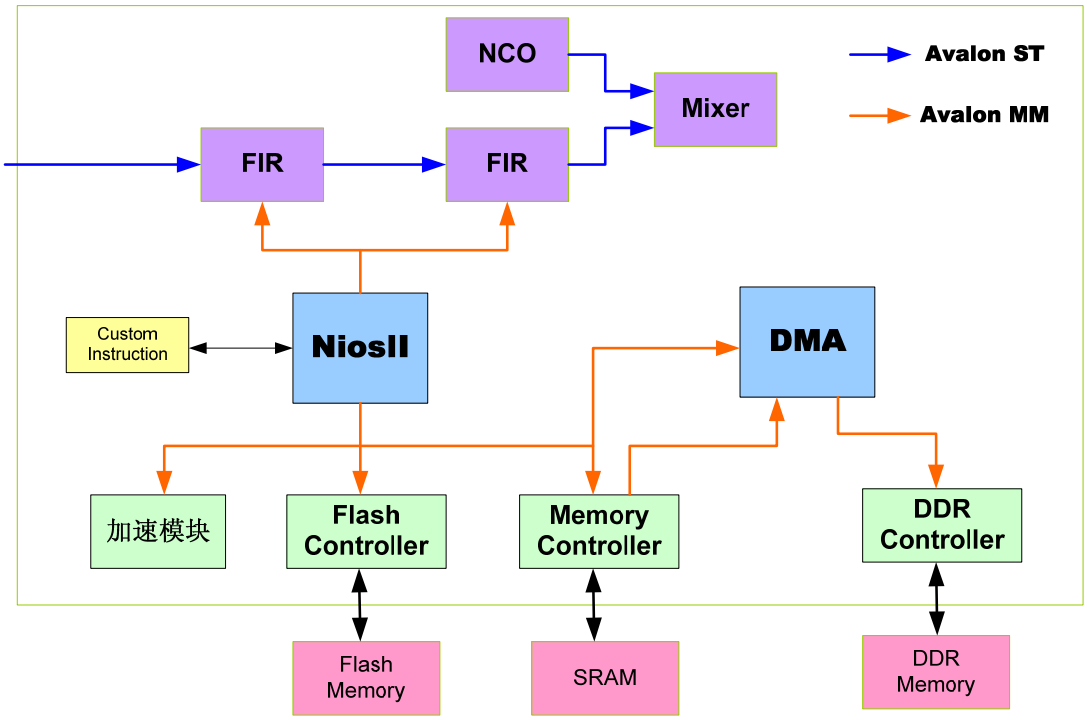
- ***数据搬运：***

现在又带来一个新的问题，如果我们需要搬运很多的数据，从一个地方到另外一个地方，怎么办呢？通过 NiosII 可能会太慢了，同时这样的事情，是不是太过无聊了一些，用 CPU 来做大材小用了。那么这个时候，我们可以用一个叫 DMA 的长工来搬箱子。



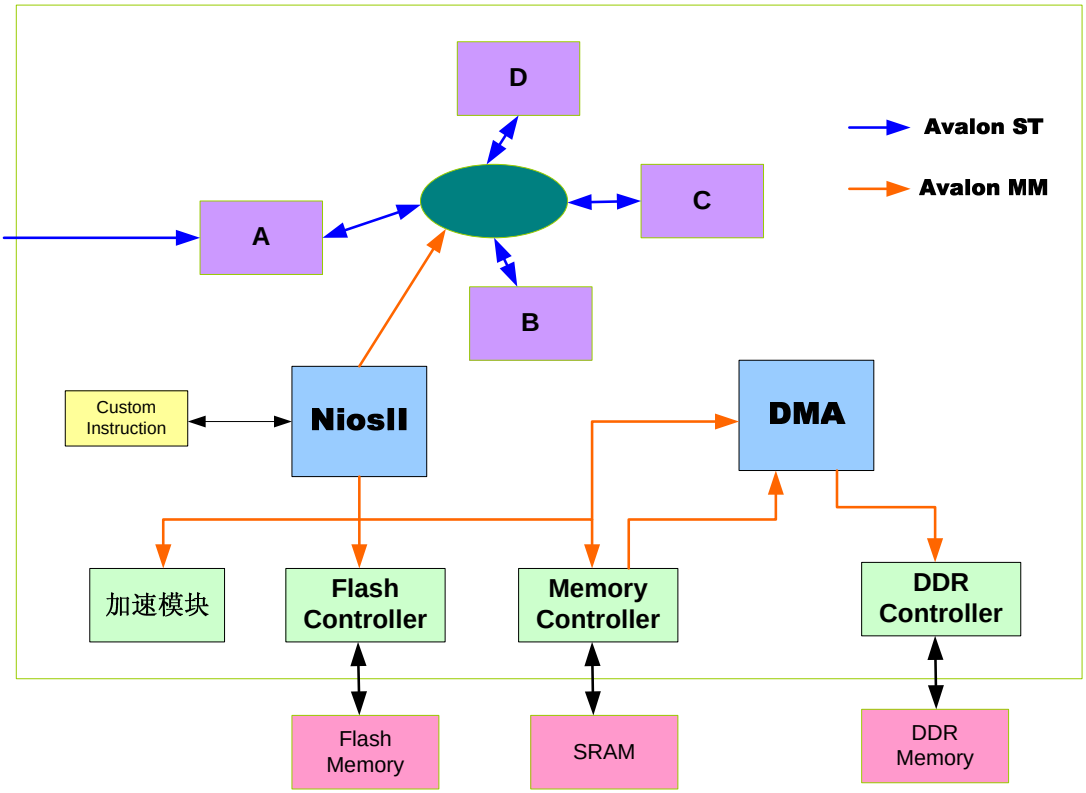
- *高吞吐量：*

现在要求更高了些了，就是我对数据的计算量要求非常大，数据又是源源不断的进来了。我们必须很快的处理数据，同时可以做相应的控制。那么我们就引用了 Avalon-ST 来提高数据的传输速度：



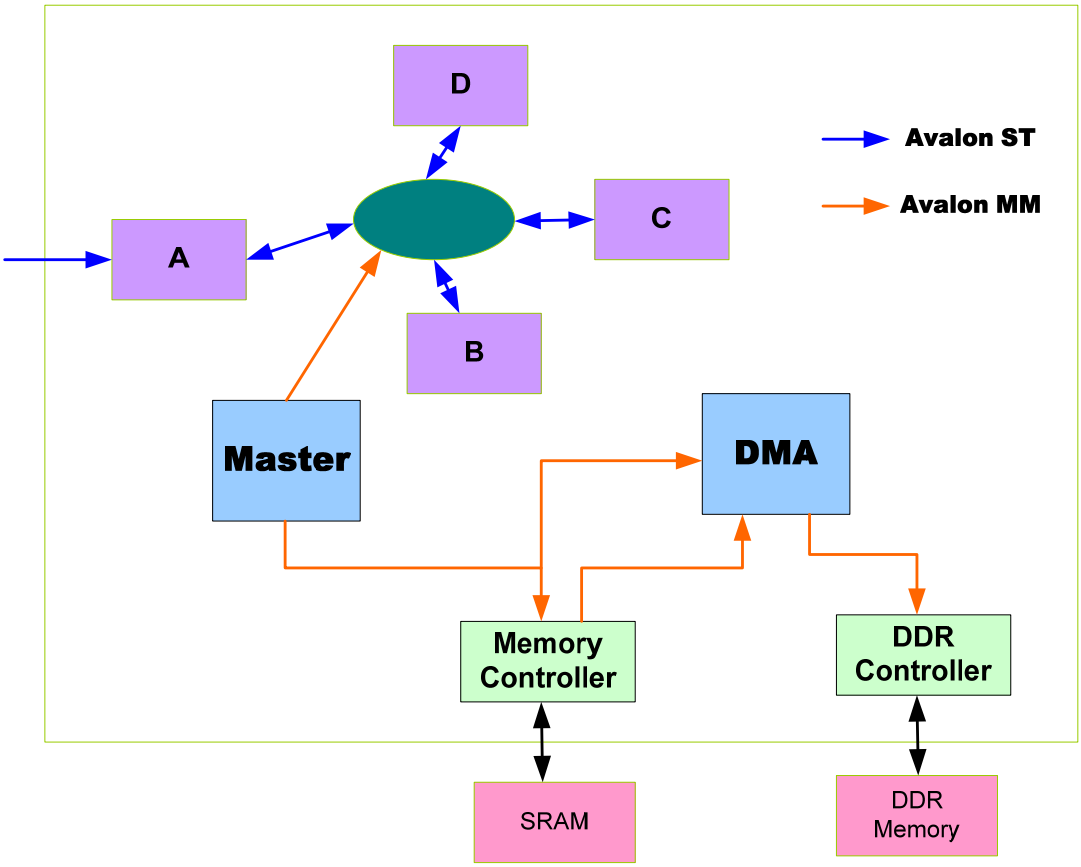
我需要更大的灵活度：

系统又提意见了，这样的设计速度是有了，可是没有灵活性啊。针对不同的协议，我可以需要的操作不一样呀，你不能这么死的连在一起。好像曹操在赤壁一样，那一把火过来，不全完了吗？好吧，那么我们可以做个桥，作为各个操作模块之间的控制。它可以根据状况来控制整个处理的流程。而每个处理模块也只是和他进行交流，所有的模块从它获得数据，然后把数据转回。



- ***一定要 NiosII 吗？***

不一定。在系统中，没有任何一样东西是必须的。只有有用的东西，没有一定的东西。而去掉一个 NiosII 的好处是可以节省很多资源，包括 FPGA 内的，和外部的。NiosII 没什么特别的能力，就是一个大型的状态机，或者说一个 Master 而已。 当然可以把它拿掉，照样还是一个 SOPC 系统



所以，系统这个东西，没有最好的，只有最适合的。你在了解自己的应用的基础上，选择你觉得最适合的方案。幸运的是 SOPC 这个平台，已经提供了你所需要的基础建设。一些具体的模块当然需要自己去做的。

大家一起来搭积木好了。