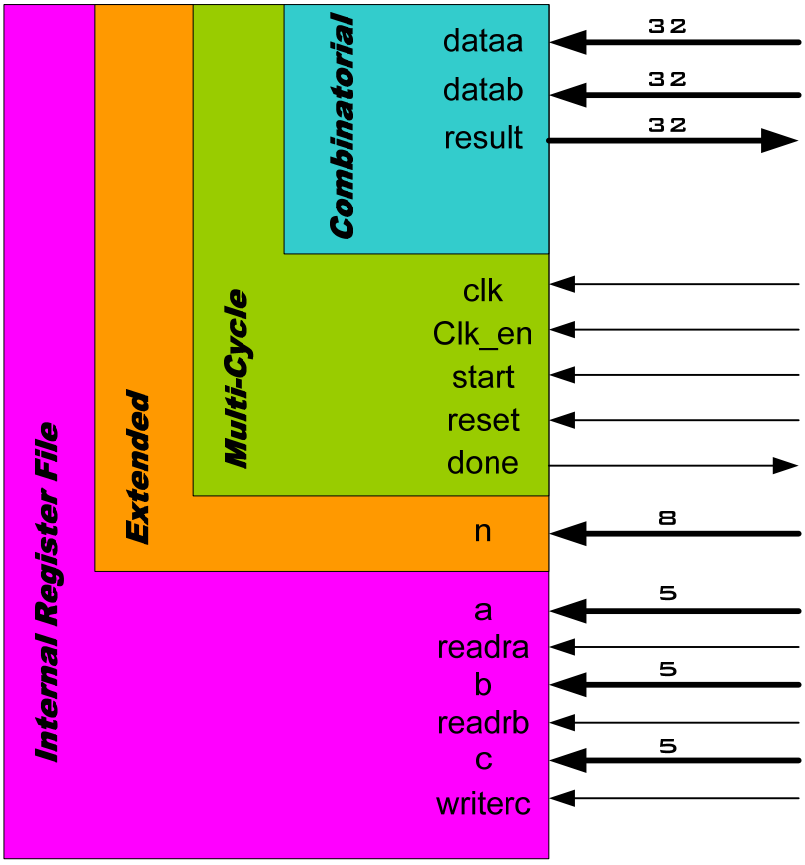
如虎添翼之 Custom Instruction

我们首先来想一个问题，我们为什么要用 NiosII 这么个 CPU？当然答案会有很多的，其中一个就是它可以在 FPGA 中实现。那么在 FPGA 中实现有什么特别的呢？那就是 FPGA 中有其他的硬件逻辑可以使用。而最大的使用这些硬件，就是 NiosII 可以获得的最大优势。其中一种方式就是使用硬件模块，而另一种方式，就是使用自定义指令。所谓的自定义指令并不是常规意义上的宏定义。而是说，我可以用硬件逻辑来实现一些功能，把这些硬件嵌入到 NiosII 中间去作为它的一个指令来使用。这样的方式，对于硬件本身的调用，资源复用，以及产品的灵活性都是有非常大的好处的。所以通过这样一种模式来使得 NiosII 本身的能力数倍的增强。

具体实现什么功能，当然要大家自己辛苦辛苦了，我这里只是介绍一些接口啊什么乱七八糟的。对大家有一些引导作用就好。至于具体的应用，还是需要自己深入了解和尝试。否则还是会玩不太好。



这就是所有 Custom instruction 需要用到的接口了。

Combinatorial:

这很容易理解，就是一些组合逻辑做出来的硬件。它需要你在一个时钟内完成操作。Dataa 和 datab 都是输入，而 result 当然就是结果了。比如说：
Result = dataa + datab
当然我们不会真的去做这么简单的逻辑，否则也不需要设计了，NiosII 自己就可以有。但是比如说， (dataa + datab + D) * E 这种呢？（D 和 E 为常数）

Multi-Cycle

一个时钟有的时候实在是做不了什么事情，所以我们可以有多时钟模式。在这种模式中有两种实现方法，一种是已经知道时钟数，或者说处理的时钟数是固定的，这只需要在参数中设置就好了。还有一种方式是处理时间不固定的，那么这个时候，我们通过 start 来开始，用 done 来表示处理结束

Extended

Extened,顾名思义，当然就是扩展方式。就是说一个自定义指令可以做不同的作用。通过 n 信号来选择需要做的具体是那种操作。当一些类似的操作，可以通过共享资源来节省空间的时候，这种模式就很有用了。而且由于 NiosII 是串行处理的 CPU, 所以在提交指令的时候，只能运行一条，而不能并行，所以同时映射非常多的自定义指令，并不能提高操作的性能。

Internal Register File

这有点复杂了，有些时候，你需要存储一些中间过程，然后再下一次操作的时候继续使用那些中间值，这就需要在硬件里面做一些存储器。而前面看到的那些操作并没有存储的功能。所以这个最复杂的模式就具有对内部寄存器的管理作用。通过 Reada, Readb, Writec 信号来控制。当 Reada 信号为低的时候，说明使用的数据为 dataa, 而当 reada 为高的时候，说明需要用到的是内部寄存器的一个值，而这个值是通过 a 为地址来找到的。同样的，当 wrtiec 为低的时候，结果直接通过 result 传出来，而为低的时候，结果会被写到 c 地址的寄存器去。千万不要来问我 a, b, c 是不是同一个寄存器的地址，因为我永远不会知道。别忘记了这点，所谓 custom instruction，是你自己做的，而这里定义的只是一些接口而已。至于说你怎么去使用这些接口，那是你自己的事情了，唯一的要求就是遵守接口的规则。

软件接口
接下来我们需要知道怎么写软件接口了。前三种方式非常简单，system.h 文件里面会自己定义好像下面这样：

```
#define ALT_CI_N 0x00
#define ALT_CI(n,A,B) __builtin_custom_ini(ALT_CI_BSWAP_N+n,(A),(B))
```

这里可以看到，当我们有 n 这个信号的时候，我们事实上把一个自定义指令扩展成为 n 个指

令。通过传递 A, B 我们可以指示输入信号。

但是对于内部寄存器模式，我们只好用汇编语言来进行一些操作。因为他需要对自定义指令内部的寄存器进行操作，我们来看下面的两端话：

Custom 0, r6, r7, r8

Custom 0, c1, r2, c4
他们使用的是同一种命令，但是结果却是完全不一样的。R 代表的是 NiosII 的内部寄存器，r6, 代表的就是 6 号寄存器。C 代表的是自定义指令的内部寄存器，c4 就是 4 号寄存器。所以第一个指令是说，我们来使用 0 号自定义指令来操作，操作的输入为 R7 和 R8, 而输出存放在 R6 里面。而第二个指令是说，我们来使用 0 号自定义指令，而操作的输入为 r2 和 c4，就是 NiosII 的 2 号寄存器和自定义指令里面的 4 号寄存器，结果放回到 C1 就是自定义指令的 1 号寄存器。通过这种方法我们来完成 internal register file 这种模式的自定义指令。