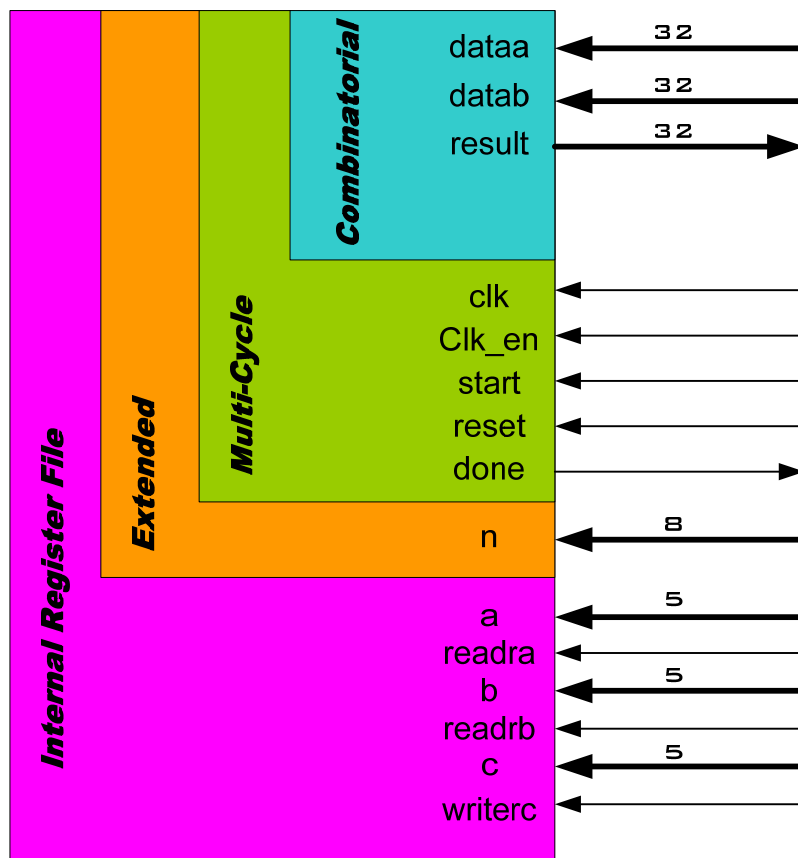




如虎添翼之 Custom Instruction

我們首先來想一個問題，我們為什麼要用 NiosII 這麼個 CPU？當然答案會有很多的，其中一個就是它可以在 FPGA 中實現。那麼在 FPGA 中實現有什麼特別的呢？那就是 FPGA 中有其他的硬體邏輯可以使用。而最大的使用這些硬體，就是 NiosII 可以獲得的最大優勢。其中一種方式就是使用硬體模組，而另一種方式，就是使用自定義指令。所謂的自定義指令並不是常規意義上的巨集定義。而是說，我可以用硬體邏輯來實現一些功能，把這些硬體嵌入到 NiosII 中間去作為它的一個指令來使用。這樣的方式，對於硬體本身的調用，資源複用，以及產品的靈活性都是有非常大的好處的。所以通過這樣一種模式來使得 NiosII 本身的能力數倍的增強。

具體實現什麼功能，當然要大家自己辛苦辛苦了，我這裏只是介紹一些介面啊什麼亂七八糟的。對大家有一些引導作用就好。至於具體的應用，還是需要自己深入瞭解和嘗試。否則還是會玩不太好。



這就是所有 Custom instruction 需要用到的介面了。

Combinatorial:

這很容易理解，就是一些組合邏輯做出來的硬體。它需要你一個時鐘內完成操作。Dataa 和 datab 都是輸入，而 result 當然就是結果了。比如說：

Result = dataa + datab

當然我們不會真的去做這麼簡單的邏輯，否則也不需要設計了，NiosII 自己就可以有。但是比如說，(dataa + datab + D) * E 這種呢？(D 和 E 為常數)

Multi-Cycle

一個時鐘有的時候實在是做不了什麼事情，所以我們可以有多時鐘模式。在這種模式中有兩種實現方法，一種是已經知道時鐘數，或者說處理的時鐘數是固定的，這只需要在參數中設置就好了。還有一種方式是處理時間不固定的，那麼這個時候，我們通過 start 來開始，用 done 來表示處理結束

Extended

Extended 顧名思義，當然就是擴展方式。就是說一個自定義指令可以做不同的作用。通過 n 信號來選擇需要做的具體是那種操作。當一些類似的操作，可以通過共用資源來節省空間的時候，這種模式就很有用了。而且由於 NiosII 是串列處理的 CPU，所以在提交指令的時候，只能運行一條，而不能並行，所以同時映射非常多的自定義指令，並不能提高操作的性能。

Internal Register File

這有點複雜了，有些時候，你需要存儲一些中間過程，然後再下一次操作的時候繼續使用那些中間值，這就需要在硬體裏面做一些記憶體。而前面看到的那些操作並沒有存儲的功能。所以這個最複雜的模式就具有對內部寄存器的管理作用。通過 Reada, Readb, Writec 信號來控制。當 Reada 信號為低的時候，說明使用的資料為 dataa，而當 reada 為高的時候，說明需要用到的是內部寄存器的一個值，而這個值是通過 a 為位址來找到的。同樣的，當 wrtiec 為低的時候，結果直接通過 result 傳出來，而為低的時候，結果會被寫到 c 位址的寄存器去。千萬不要來問我 a, b, c 是不是同一個寄存器的位址，因為我永遠不會知道。別忘記了這點，所謂 custom instruction，是你自己做的，而這裏定義的只是一些介面而已。至於說你怎麼去使用這些介面，那是你自己的事情了，唯一的要求就是遵守介面的規則。

軟體介面

接下來我們需要知道怎麼寫軟體介面了。前三種方式非常簡單，system.h 檔裏面會自己定義好像下面這樣：

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

這裏可以看到，當我們有 n 這個信號的時候，我們事實上把一個自定義指令擴展成爲 n 個指

令。通過傳遞 A, B 我們可以指示輸入信號。

但是對於內部寄存器模式，我們只好用組合語言來進行一些操作。因為他需要對自定義指令內部的寄存器進行操作，我們來看下面的兩端話：

Custom 0, r6, r7, r8

Custom 0, c1, r2, c4

他們使用的是同一種命令，但是結果卻是完全不一樣的。R 代表的是 NiosII 的內部寄存器，r6，代表的就是 6 號寄存器。C 代表的是自定義指令的內部寄存器，c4 就是 4 號寄存器。所以第一個指令是說，我們來使用 0 號自定義指令來操作，操作的輸入為 R7 和 R8，而輸出存放在 R6 裏面。而第二個指令是說，我們來使用 0 號自定義指令，而操作的輸入為 r2 和 c4，就是 NiosII 的 2 號寄存器和自定義指令裏面的 4 號寄存器，結果放回到 C1 就是自定義指令的 1 號寄存器。通過這種方法我們來完成 internal register file 這種模式的自定義指令。