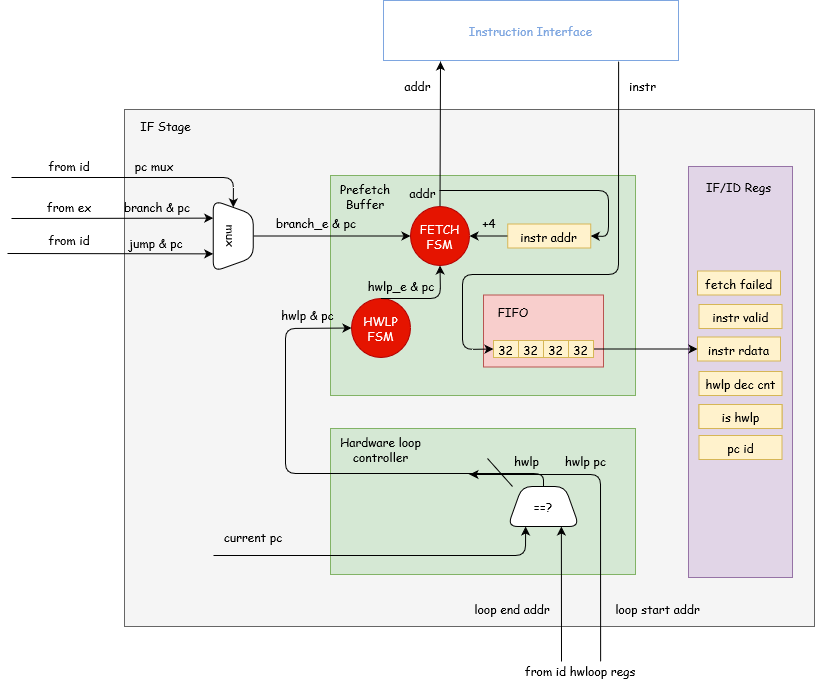
Pulp RI5CY代码详解

前言：在阅读Pulp RI5CY本身手册（项目代码doc目录下user manual）后，只能对RI5CY的特性有一些了解。在此基础上，想要看懂RI5CY的代码，还需要一番解释。

1. **IF段代码详解**

**重点有：iMem Interface，Prefetch Buffer，Hardware Loop（hwlp）**

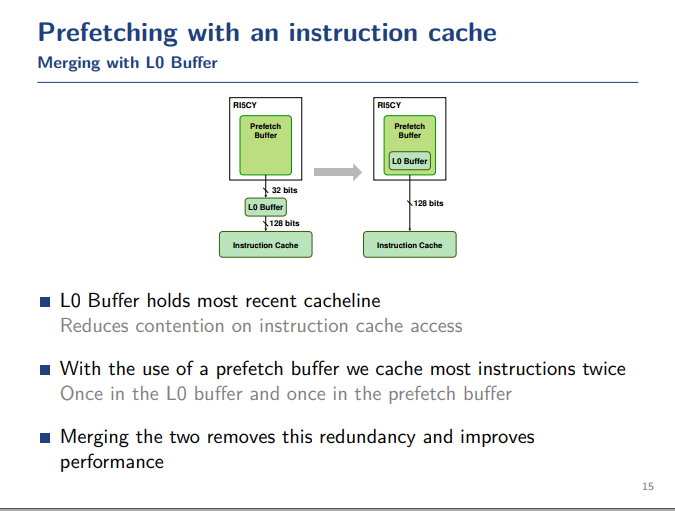


**文件关系：**

IF段的例化是在文件riscv core中，想要了解IF段信号来源去向，要查riscv core。IF段代码文件是riscv if stage，其中包括一些简单的组合电路、指令预译码的例化、Prefetch Buffer的例化、hwlp的例化和IF/ID段寄存器。

**指令预译码**代码文件是riscv compressed decoder，主要是要得到两个信息，其一是该指令是否为压缩指令，因为压缩指令是16位的，要特殊处理；另一是该指令是否为非法指令，若指令非法，则产生非法指令信号。

**Prefetch Buffer**可以根据参数RDATA\_WIDTH是32还是128有两种配置方式，分别对应代码文件riscv prefetch buffer和riscv prefetch L0 buffer。Prefetch Buffer的目的我也无意中看到，见下图（PULP在2015年RISCV大会上的PPT）。我目前研究默认情况，即RDATA\_WIDTH为32的情况，该prefetch buffer主要包括两个FSM和一个指令FIFO，一个FSM用来处理与指令接口交互（即取指）；另一个FSM用来处理Hardware Loop的跳转；而FIFO则是由4个32位寄存器构成，用来存储预取的指令。



Hwlp代码文件是riscv hwloop controller，主要通过读取位于riscv id stage文件中的hwloop reg和当前的pc来判断是否要进行一个硬件循环。

不妨首先介绍它的hwlp。

**HWLP要求**：

1. 循环体至少有两条指令；
2. 至少循环一次，即类似do while循环模式。

参考user manual可知，hwlp硬件特性的使用要通过其提供的ISA扩展，可参考user manual 14.2。简单说来就是通过自定义指令设置hwlp的起始地址、结束地址和循环计数器。而RI5CY最多支持两层循环嵌套，在其硬件实现中，可观察到代码中参数N\_HWLP的默认值是2，相互印证。

硬件循环的具体实现是，通过每周期比较PC值和循环结束地址，若相等，即当前已经执行到一个循环的末尾，则根据counter判断是否需要循环。如果需要，则生成hwloop使能，并将循环开始地址送出作为跳转地址，且生成counter递减信号，表示已经执行一次循环体。

**iMEM Interface：**

RI5CY的内存接口在user manual的第2、3章中描述很详细，总结一下就是请求应答式的访问。读iMEM要：

1. 请求，发送req和addr，等待gnt回复，表示内存已经收到请求。该过程cpu主动；
2. 应答，收到rvalid和rdata。该过程imem主动。

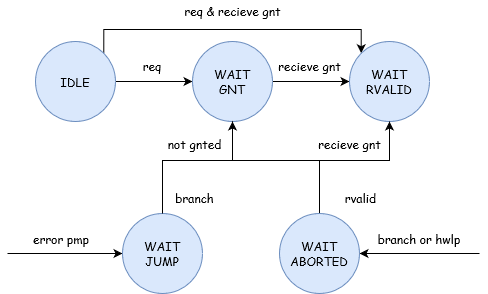
如果一直未收到rvalid，RI5CY会一直等待。且请求与应答中间可插入下个请求，顺序请求，顺序应答。

通过该接口对iMEM进行访问则是在Prefetch Buffer中通过FSM来实现。

**Prefetch Buffer：**

所有取到的指令一次存入FIFO，而FIFO则给ID段提供指令。

访问iMEM的FSM如下图：



如图所示，系统复位后FSM默认为IDLE状态。一般过程是：

1. 收到指令请求req，开始读指令，发送instr addr，进入WAIT GNT状态，等待内存回复；
2. 收到内存回复gnt，进入WAIT RVALID状态，等待指令；
3. 收到rvalid，完成读指令。

但是因为：

1. 流水线要每个周期都读指令：
   1. 在WAIT RVALID状态仍然可以发送请求；
   2. 发送请求后，如果收到gnt，停在WAIT RVALID状态；
   3. 否则，停在WAIT GNT状态。
2. 流水线存在跳转指令和硬件循环：
   1. 如果发生跳转和硬件循环，则当前取指并非流水线需要的指令，进入WAIT ABORTED状态，等待当前取指的应答。
   2. 收到之前请求的应答后，重新开始取指过程。
3. 流水线可能会非法访问内存：
   1. 收到error pmp信号后，说明流水线发生非法访问，进入WAIT JUMP状态等待处理；
   2. 收到跳转信号，说明处理开始，重新取指。
   3. （这里未搞清楚pmp error的信息由谁来处理，新的跳转信号如何产生）

在正常的指令流中，PC一直加4进行取指，如发生跳转则要对新地址进行取指。而HWLOOP FSM则负责处理这两种情况下发生HWLOOP时的情景。该问题复杂在，若当前指令是HWLOOP最后一条指令，但是流水线中已有跳转指令未执行完，要先处理跳转指令。该部分参照代码和上面描述容易理解。

1. **ID段代码详解**