CoefLoadUnit模块被拆分为三个模块，分别为WaveLoadUnit，TimeShiftLoadUnit，DistanceLoadUnit。

折叠因子有radius factor和depth factor，简写r和d。暂时假设目前的设计只考虑一个频率的情况。

1. WaveLoadUnit

Wave矩阵有一共 R\*D 个元素，需要的reg数量是 R\*D\*32 bit。我这里考虑的设计是将总的存储空间划分为D个，每个存储空间可以是寄存器组或者是SRAM。无论是哪种，输入都是串行的，均通过AXI4总线结构将外部的wave数据输入存储空间。

如果考虑的是全部使用寄存器的设计，面积消耗可能会很大，同时寄存器的排列是不规整的，但可以串行输入并行输出，缩短了wave挂载到相应位置所需要的时间，(r+1)cycles。而如果采用的是SRAM的设计，面积消耗小，排列规整，串行输入串行输出，(r+r)cycles。

控制信号：

* Ready\_for\_store: 当内部存储空间所存储的所有数据均被消耗/使用过了，此信号拉低，用来接受总线上的数据时拉高。
* Data\_enable: 当总线将内部存储空间全部填满的时候，拉高此信号，表示此时所有数据均已准备就绪，可以挂载在相应的端口上了。与axi 的last信号不同，这里这个信号值的含义是存储块内的所有数据均已load完成。

1. TimeShiftLoadUnit

Timeshift向量一共有D个元素，需要的reg数量是D\*32 bit，对于每个rsd kernel core来说应该比较小的，可以使用全部寄存器的设计。

1. DistanceLoadUnit

Distance向量一共有D个元素，需要的reg数量是D\*32 bit，情况同上。

以上三个模块的data\_enable信号均拉高时，start信号拉高，指示impulse可以开始输出，