文件夹结构：smartnic文件夹是实验室在cordum（祖师爷）代码的基础上进一步做工作搭建的智能网卡领域的各种项目的文件夹。其中各个路径都是不同项目对应的程序代码。

以DDR读写工程为例介绍工程文件结构以及原理：

我们自己写的axi\_ddr\_capacity\_test.v文件是最顶层的模块，由于读写DDR非常复杂，因此需要借助框架来对DDR读写，这个框架是mig核，是实验室为了完成DDR读写而搭建的现成框架。Mig框架完成的任务是一端接axi协议接口，一端接DDR接口（使用mig.prj来完成），整个工程由外层到内层由多个verilog文件层层调用（对应是外层内层模块层层嵌套），最外层是FPGA.v，FPGA.v文件是根据整个实际板卡的硬件结构来写的，是一个非常全面的文件。FPGA.v模块文件的输入输出信号通过xdc文件与FPGA板子上的管脚相绑定，这样就完成了虚拟的代码与实际电路的对应，板卡就可以识别出来代码说的是什么。xdc文件本质上讲需要将最底层的 verilog文件中的输入输出信号与管脚全部绑定，但是由于DDR太难，使用prj的方式绑定axi与mig，使用mig去配置DDR管脚。因此在这个xdc文件中只需要帮剩余信号clk与rst与led，而这些也在现有框架内绑好了。并通过内外层模块嵌套、例化的方式去把对板子管脚的控制送到最内层自己写的模块信号线上。

如果要从零开始去写一个小模块，自己去按需求使用板卡上的一些资源，自己写一个xdc文件绑定物理管脚，就相当于自己写了一个小框架，然后自己写的小模块（只有一层）就是最外层的模块。

下一步要把xdc文件怎么写的，里面的语句如何理解搞明白

把多层文件之间信号的来龙去脉搞明白

ILA是vivado中自带的一种监听信号的工具，可以采样一定拍数（时间）内的信号线的值 ，在实验室项目开发中的使用方法是将ila.tcl文件加入ip文件夹中，相当于加入了这个ip，然后在对应工程的makefile中加入调用该ip的语句。之后按照ila.tcl文件中的例化方式，去在对应想要监听的信号所在模块文件内部进行例化。

跨时钟域操作，是vivado中自带的一个功能，使用这个操作可以将与模块时钟频率不一样的信号转换为一个新的信号，从而将这个新的信号接近模块内部，这个直接在需要跨时钟域信号所在模块例化就可以了，将转换后的信号定义为一个中间信号，将这个信号引入需要该信号的模块即可。

将这些做完之后，在makefile文件路径下编译，然后打开xpr工程文件，打开implement design ，在tcl console中使用命令在工程编译路径下生成一个ltx文件（debug文件）然后在烧录文件的时候第一行加入bit文件、第二行加入该文件即可。