1.指令系统（ISA）的定义和实现

两个指令机例子：MIPS和RISC-V（后者更新一点，资料也多一点）

2.关注怎样提高指令系统的性能，比如：

单流水线指令集（ILP）（细粒度），指令前瞻（Speculation）

3.一条指令可同时处理多个数据，靠存储器的带宽不断增加实现（DLP），UPU→GPU/SPE OpenCL/CUDA

4.处理器里面的晶体管不断增加，把单核拆成比较小的多核（Course-grained）（粗粒度）：

线程级并行TLP（需要编程模型进行编程规范，例如用于多核的OpenMP，多CPU的MPI，或两者混合）,进程级并行（也叫任务级并行）PLP/Task-level——多处理机

5.DSA

本课只关注指令集系统的实现和指令集并行的开发，也就是上面的1和2

本课需要自己学习RISC-V（有人证明能在短时间内学会这个=\_=，真的不是被逼出来的吗）

需要完成：

1. RISC-V集 I-32/16 + 10条扩展指令（自定义）
2. Cache 分离的L1D+L1I
3. 指令集并行开发 只要实现动态分支预测即可

第一周：给出指令集的定义

第二周：实现指令集（不少于5级流水线）

第三周和第四周：给指令集加Cache

第五周和第六周：做动态分析预测

如果第二周没做完，再给一周时间，但不会超过三周，没做完者当场判定成绩，不会超过C

建议2-3人一个组。

用硬件描述语言，或敏捷开发语言

注意：代码会查重！

本周六（6.12）晚22：00前提交分组名单于 [lishen@nudt.edu.cn](mailto:lishen@nudt.edu.cn) 作为选课确认

下周三（6.16）12：00前完成指令系统（ISA）定义以及相关文档（主要包括指令，功能，数据类型，寄存器定义，指令格式，指令功能描述）建议参考教材上的例子。

教材包括了自己所有能找到的相关资料（废话），可以找计算机体系结构量化研究方法第六版：

