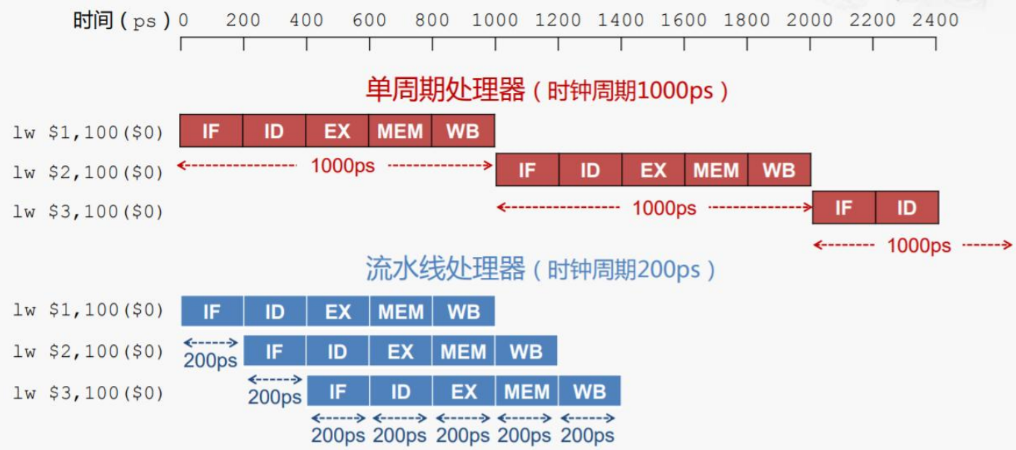
# 作业1.1

1问题： 为什么流水线前传可以提高cpu效率

1 流水线并行： 为提高cpu速度，引入了流水线，一个大的计算任务被分为多个小任务，不同任务之间可以并行。例如：取指，译码，执行访存，写回。同时可以运行多个流水线的不同步骤。



2 指令前递：一个问题是，如果第一条指令load a，第二条指令Add a。在流水线并行的情况下， load还没完成的时候，add可能开始了，形成数据依赖，这时需要将流水线暂停，串行执行指令，这叫指令前递。

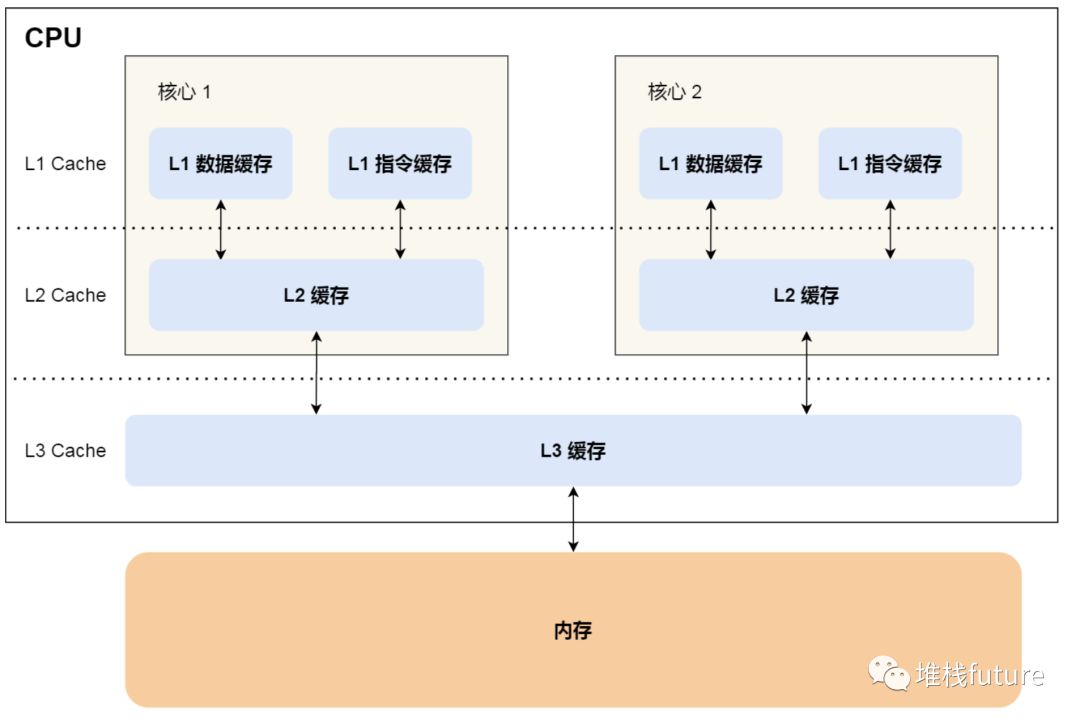
3 分支预测： cpu执行还可以根据使用的时候根据执行历史预测出分支控制的case，假设if 条件总是成立，那么就可以总是先把if条件下的数据提前加载，加快执行。如果猜错了，重新执行。

4 指令乱序，如果指令之间存在依赖，那么cpu就只能串行执行，我们可以先执行那些独立的执行，这样提高了cpu并行度，加快了执行速度。

2问题：3级缓存特点分别是什么， 分别适合什么数据

为了弥补 CPU 与内存两者之间的性能差异，就在 CPU 内部引入了 CPU Cache，也称高速缓存

CPU Cache 通常分为大小不等的三级缓存，分别是 **L1 Cache**、**L2 Cache** 和 **L3 Cache**。

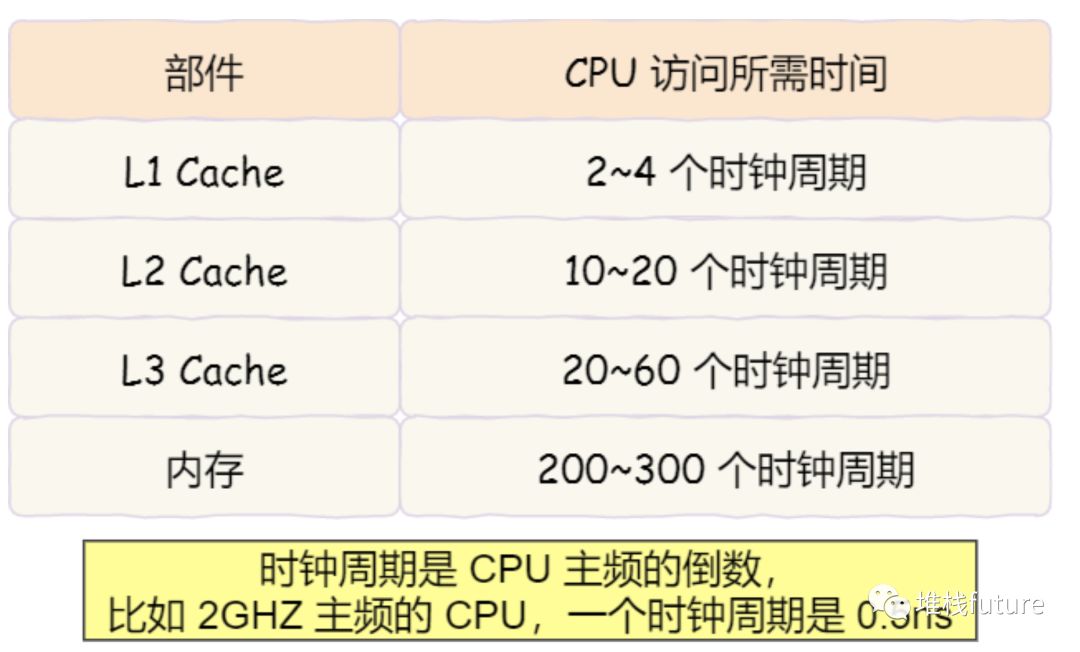


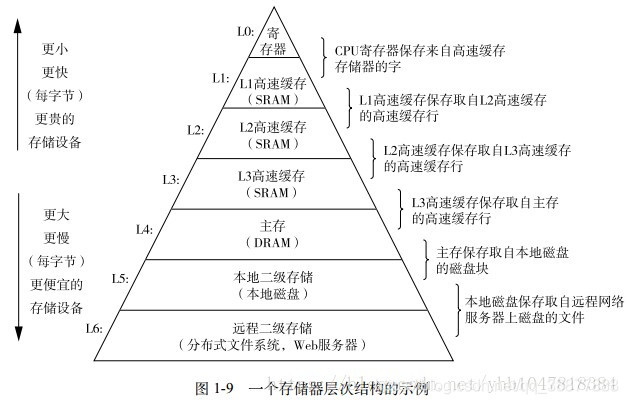
越靠近cpu核心，访问速度越快，其中L3是多个核心共享的。

L1 cache 是cpu最近的缓存，成本最高，速度最快，一般L1 Cache都是分立Cache，分为数据缓存和指令缓存，可以减少访存冲突引起的结构冒险，这样多条指令可以并行执行；内置；其成本最高，对CPU的性能影响最大  
多级Cache的情况下，L1 Cache的命中时间比命中率更重要

L2 cache 各个核心独立，容量直接影响cpu性能，一般L2 Cache都是联合Cache，这样空间利用率高  
没有L3 Cache的情况下，L2 Cache的命中率比命中时间更重要（缺失时需从主存取数，并要送L1和L2 cache）

L3 cache 进一步降低内存访问延迟，是核心间共享的，l3 cache多为外置，由静态存储芯片SRAM组成，容量较小但比主存DRAM技术更加昂贵而快速， 接近于CPU的速度。CPU往往需要重复读取同样的数据块， Cache的引入与缓存容量的增大，可以大幅提升CPU内部读取数据的命中率，从而提高系统性能





总结： l3 cache缓存 主存内容，一次一个cache line为单位，l2 cache缓存l3数据，core内使用，l1 cache缓存 l2 cache， 区分数据和指令。由运算单元直接使用。

3 问题：什么问题适合gpu

gpu 吞吐优先

1 对于计算密集型任务适合使用gpu进行计算，涉及网络，存储等任务（io操作）不适合

2 对于计算任务可以清晰拆解为多个小任务，无复杂控制流程。

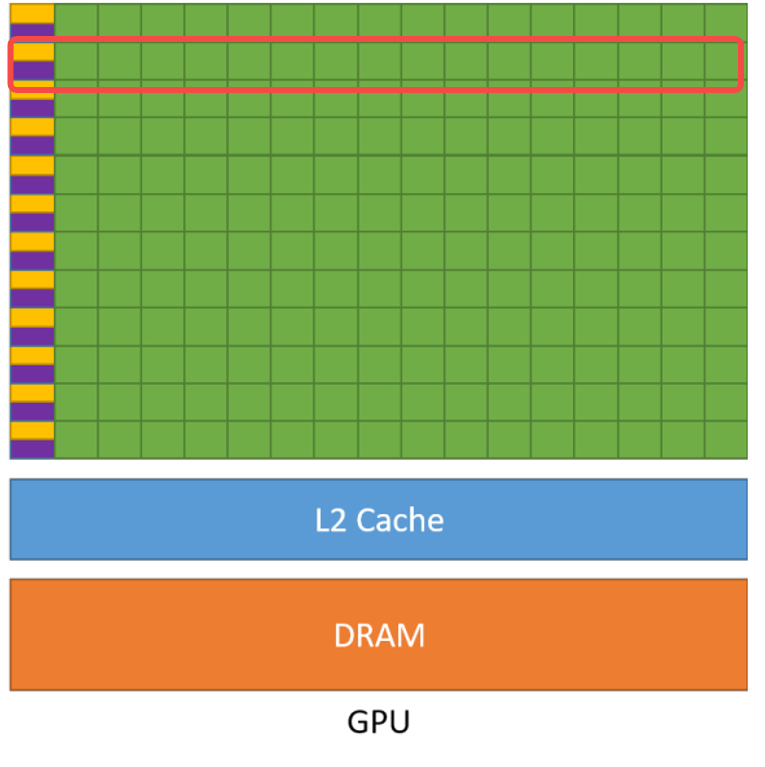
# 作业1.2

1 gpu中控制和计算的单元如何结合/warp如何被执行

一个sm中多个线程并行执行，他们被以warp为单位进行调度和执行。Warp是基本执行单元，一个warp 32 和并行thread，smit模式，所有的thread执行同一个指令，每个thread使用自己的data进行计算。

一个warp中thread必然在一个block中，如果一个block中thread数不是warp的整数，那么会有一个warp有一些inactivate的线程。

如作业中图中红框，一行共享一个控制单元，一行中每个cell是一个硬件线程，他们执行相同的就算指令。



2 为什么说线程warp是执行单位

接着第一个问题的回答，虽然一个warp中thread执行相同的指令，但是每个thread执行不同的数据，Warp是硬件同时执行的单位。只能以这个单位进行调度和执行。

3 tread(3,0,0) 的唯一 id

如下图，

1 grid块数是2 \* 2, 每个block 16个thread， 两维，thread的block id是（1，1）,

于是： y \* blockdim.x + x的到: 16\*2\* 1 + 16 = 48

2一个block是三维的 2 \* 4 \* 2， 若(3,0,0)为x=3, y= 0, z=0, -> 0 \* 2 \* 4 + 0 \* 2 + 3

3 最终 thread 的唯一id是 = 48 + 3 = 51

