????????????????

难点、不适合

1. 难以做复杂的算法（调度、分析、统计）
2. 难以做逻辑深度比较深的计算
3. latency的长和不确定，模块间的时间不确定性
    1. 需要大量实现 “异步 ”逻辑
        1. 消耗大量的面积
    2. 需要很大的 buffer存储用于状态 footprint
        1. 消耗大量的面积
    3. 增加外部控制的复杂度
        1. PC需要对所有的操作异步化
    4. 降低实时性
4. 为了高频、增加很多级流水线
    1. 浪费大量用于同步的逻辑
    2. 容易撞到功耗墙
    3. latency的延长和不确定性

优势、适合

1. 并行任务非常适合，适合简单的、重复的、大量的计算，比如 MMU
2. 带宽足够，可以做大量的并发处理
3. 实时性好，可以做到 cycle级别的调度
4. 利用数字电路表达的特点加速
    1. 对 2的幂次的乘除运算
    2. 固定算法的 hash， mmu等处理
    3. 查找表计算: cuda的 LOP3.LUT指令
        1. 通过软件的静态预先计算，最大化压缩硬件需要处理的表达空间

各种硬件 (比如 roq， sram读写口等)的数量是有限制的，不像软件视角不感知这个冲突。

各种操作，运算都是有延迟的，都不是实时单拍完成的

微架构的各种规格，在早期设计的时候，包括建模实现，需要参数化，可配置

Revision #4
Created 2025-08-31 05:54:23 UTC by Colin
Updated 2026-04-29 07:33:40 UTC by Colin