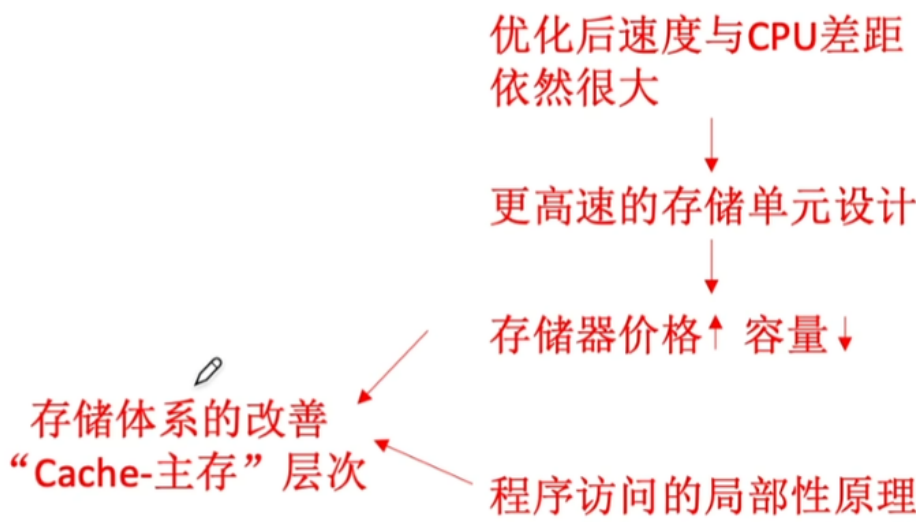


程序访问的局部性原理

存储系统存在的问题

双端口RAM、多模块存储器提高存储器的工作速度



局部性原理

哔哩哔哩

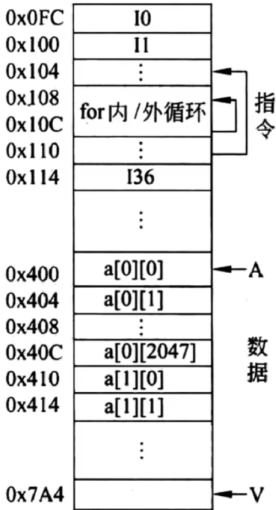
局部性原理

程序A:

```
1 int sumarrayrows(int a[M][N])
2 {
3     int i, j, sum = 0;
4     for (i = 0; i < M; i++)
5         for (j = 0; j < N; j++)
6             sum += a[i][j];
7     return sum;
8 }
```

程序B:

```
1 int sumarraycols(int a[M][N])
2 {
3     int i, j, sum = 0;
4     for (j = 0; j < N; j++)
5         for (i = 0; i < M; i++)
6             sum += a[i][j];
7     return sum;
8 }
```



指令和数据在内存中的存储

Eg: 数组元素、顺序执行的指令代码

空间局部性: 在最近的未来要用到的信息(指令和数据), 很可能与现在正在使用的信息在存储空间上是邻近的

时间局部性: 在最近的未来要用到的信息, 很可能是现在正在使用的信息

Eg: 循环结构的指令代码

基于局部性原理, 不难想到, 可以把CPU目前访问的地址“周围”的部分数据放到Cache中

程序B按“列优先”访问二维数组, 空间局部性更差