本科毕业答辩

基于深度学习的数据预取质量优化算法设计与实现

答辩人：李芳达
专业：计算机科学与技术
指导老师：蔡旻

北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

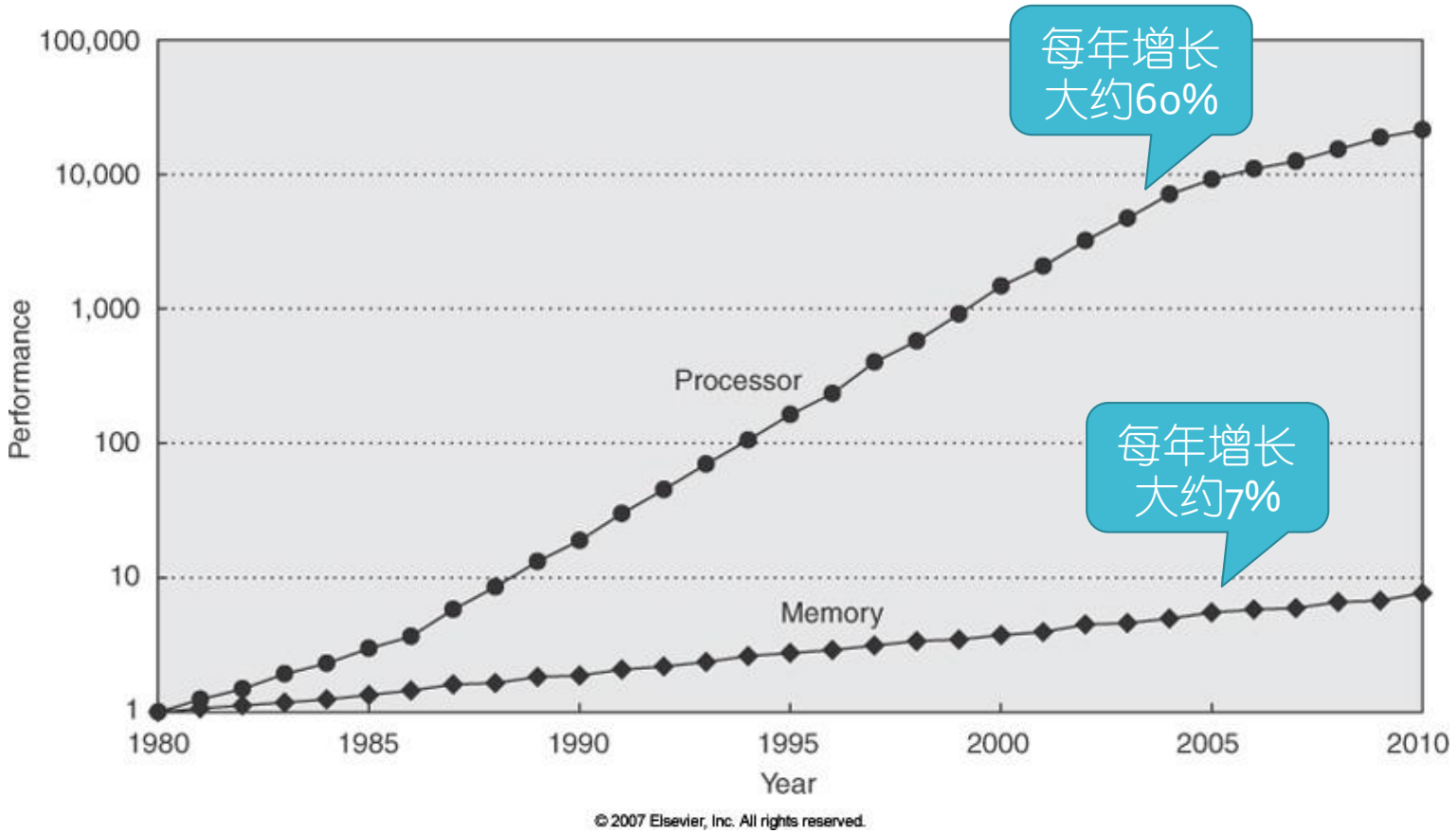
目录

北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

基于深度学习的数据预取质量优化算法设计与实现

课题背景

CPU与内存之间的速度差距逐年增大，需要找到降低或隐藏内存访问延迟的方法



北京工业大学
BEIJING UNIVERSITY OF TECHNOLOGY

基于深度学习的数据预取质量优化算法设计与实现

课题背景

解决方案	优点	缺点
cache	访问速度基本满足CPU发出的请求频率	价格数倍昂贵于传统内存 在不符合时间及空间局部性的程序上利用效率低
prefetch	降低cache缺失率 在CPU进行运算操作时并行预取	可能产生缓存污染

基于深度学习的数据预取质量优化算法设计与实现

课题背景

解决方案	优点	缺点
cache	访问速度基本满足CPU发出的请求频率	价格数倍昂贵于传统内存 在不符合时间及空间局部性的程序上利用效率低
prefetch	降低cache缺失率 在CPU进行运算操作时并行预取	可能产生缓存污染

本文的研究方向

基于深度学习的数据预取质量优化算法设计与实现

系统总体设计

开发工具

Chisel3(Constructing Hardware In a Scala Embedded Language)

--由加州大学伯克利分校的计算机架构研究小组开发

--在高级编程语言Scala的基础上设计开发的一套硬件构建语言

特点：

- 抽象数据类型和接口；
- 面向对象编程和函数构建；
- 使用高度参数化的元编程（metaprogramming）；
- 自动生成可以在标准 ASIC或 FPGA上使用的 Verilog程序 。

基于深度学习的数据预取质量优化算法设计与实现

系统总体设计

开发工具

Chisel3与Verilog比较

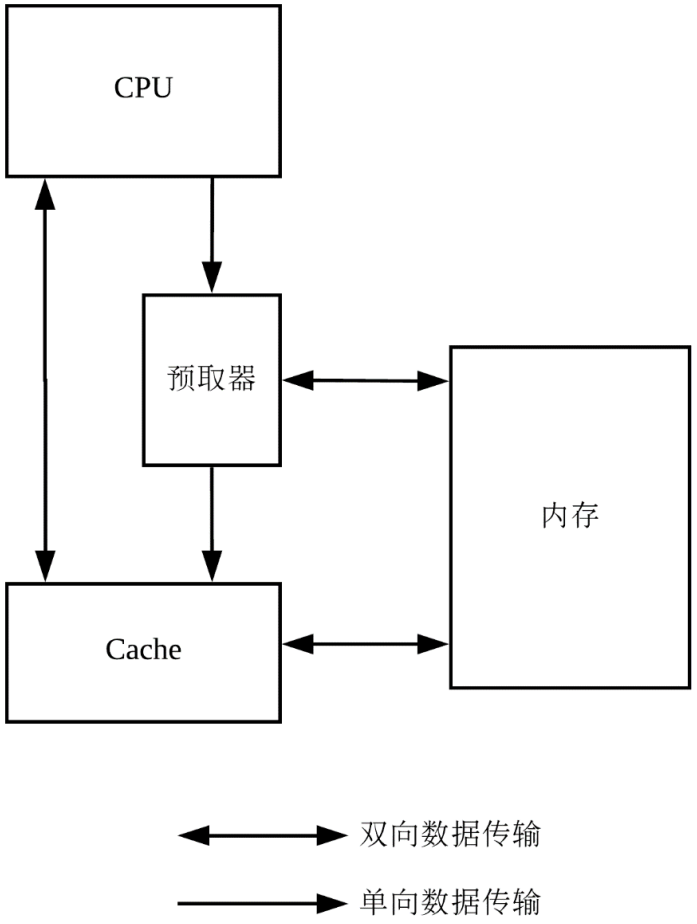
Chisel3更加直观、具有更高的设计抽象级别，可以编写电路生成器而Verilog只支持直接编写具体电路。

高级编程语言功能	Verilog	Chisel3
面向对象编程	×	√
类型推断	×	√
函数式编程	×	√
映像	×	√

基于深度学习的数据预取质量优化算法设计与实现

系统总体设计

- 模块划分



北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

基于深度学习的数据预取质量优化算法设计与实现

系统总体设计

- 全局参数定义

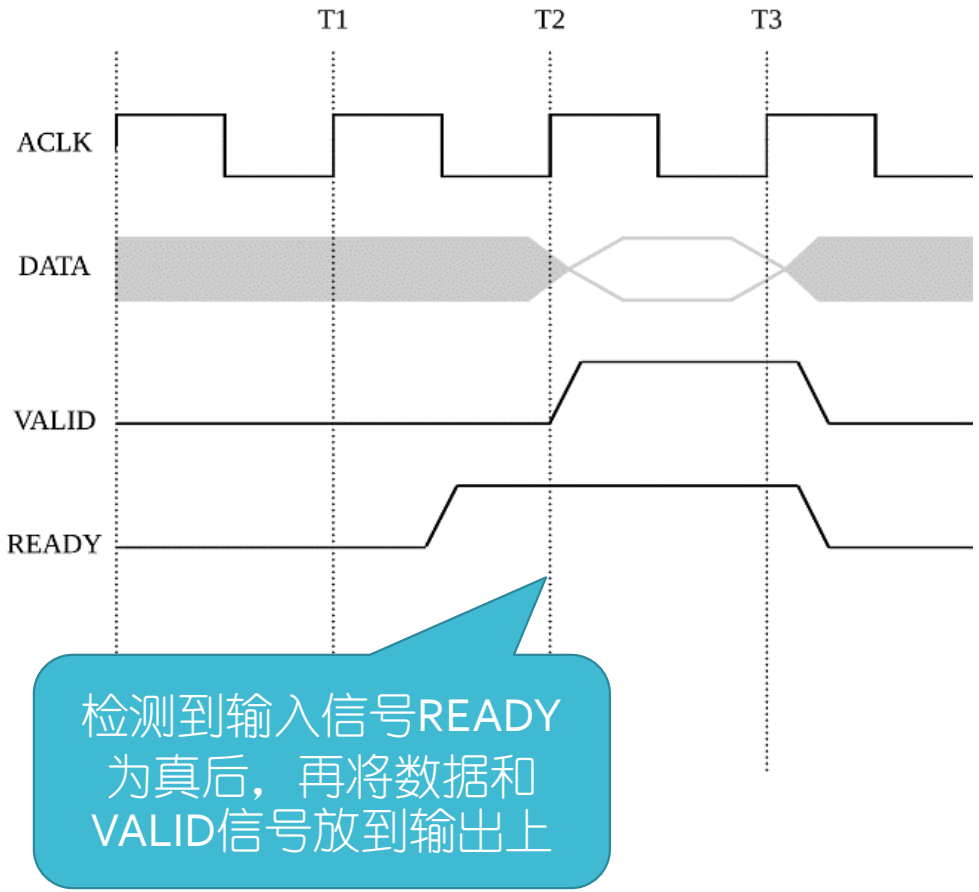
参数名称	描述
addrWidth	内存地址宽度
memSize	内存容量
cacheSize	Cache容量
blockSize	缓存块大小
assoc	Cache相联度
numSets	Cache组数

基于深度学习的数据预取质量优化算法设计与实现

系统总体设计

• 传输协议设计

主从之间的握手机制



Chisel3提供的接口类型：

DecoupledIO

- READY 输入
- VALID 输出
- DATA 输出

ValidIO

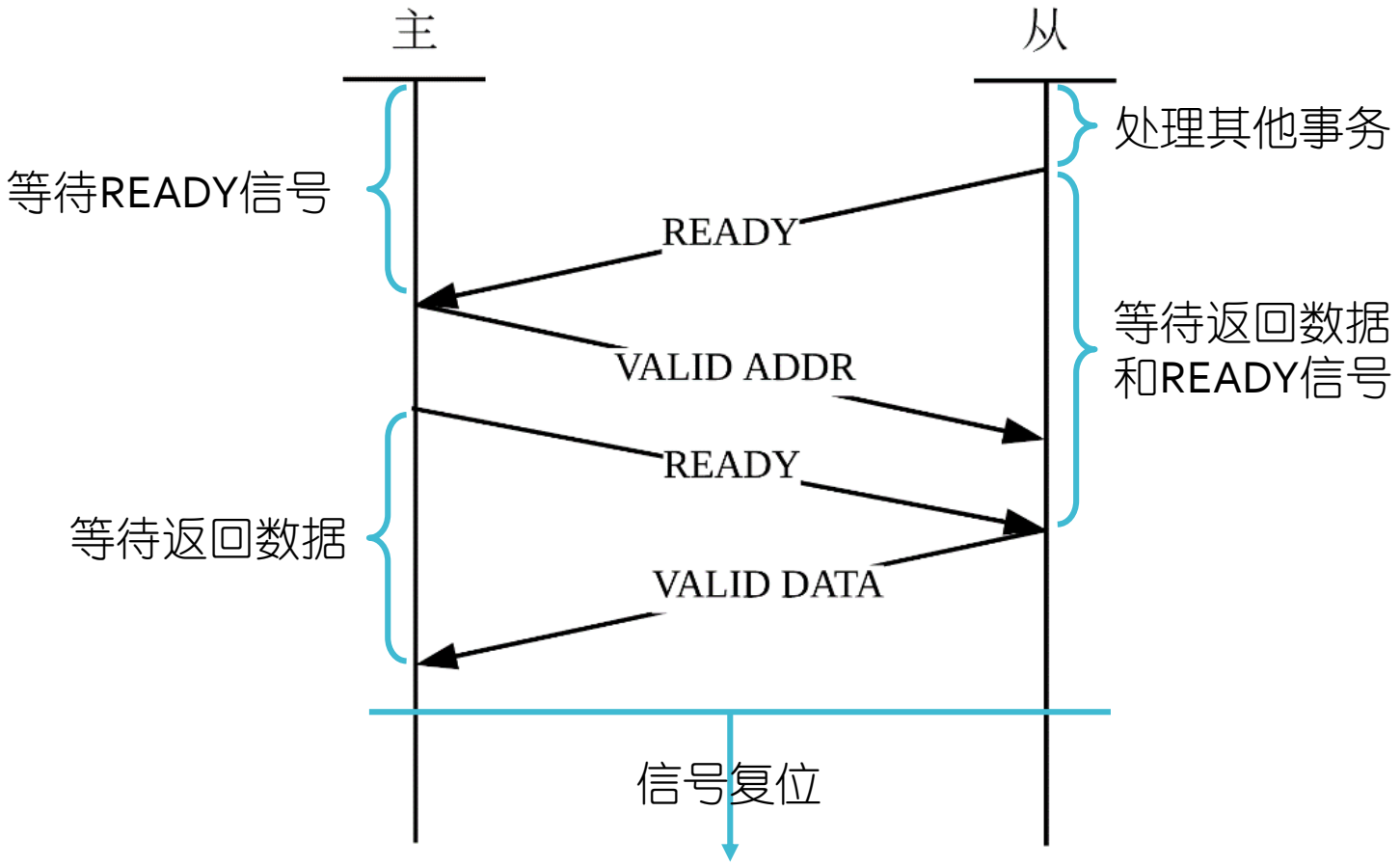
- VALID 输出
- DATA 输出

基于深度学习的数据预取质量优化算法设计与实现

系统总体设计

• 传输协议设计

主从之间的握手机制



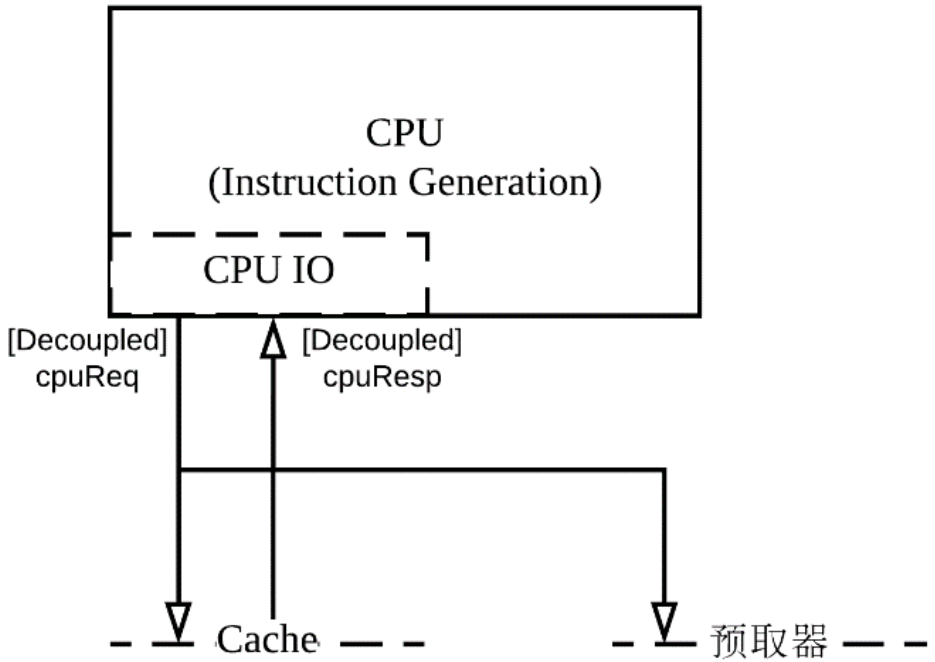
北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

基于深度学习的数据预取质量优化算法设计与实现

系统总体设计

CPU访存流生成功能

本系统中不包含具体的CPU硬件模块，但需要完成CPU对cache和预取器发出正常指令并接收cache应答的功能。



北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

基于深度学习的数据预取质量优化算法设计与实现

系统总体设计

单元测试方法

Chisel3 Testing原理：调用ScalaTest库

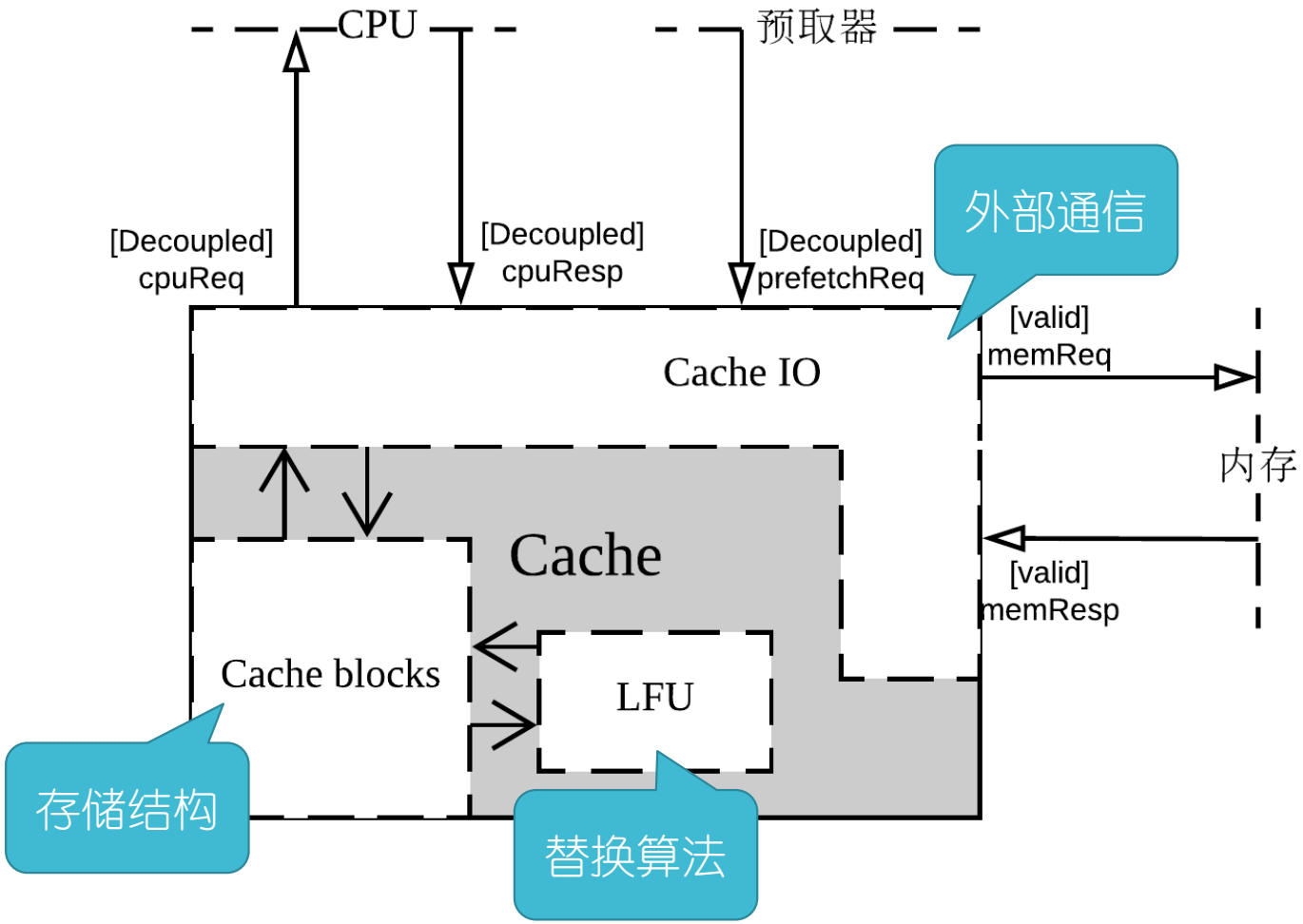
Chisel3 StandardTester 提供的基本操作：

- 复位操作：reset([n: Int])
- 单步操作：step[n: Int]
- 置数操作：poke(data: bits, x: BigInt)
- 窥探操作：peek(data: bits): BigInt
- 期待操作：expect(good: Boolean, msg: String): Boolean
  expect(data: bits, target: BigInt): Boolean

基于深度学习的数据预取质量优化算法设计与实现

存储模块的设计与实现

- Cache模块设计

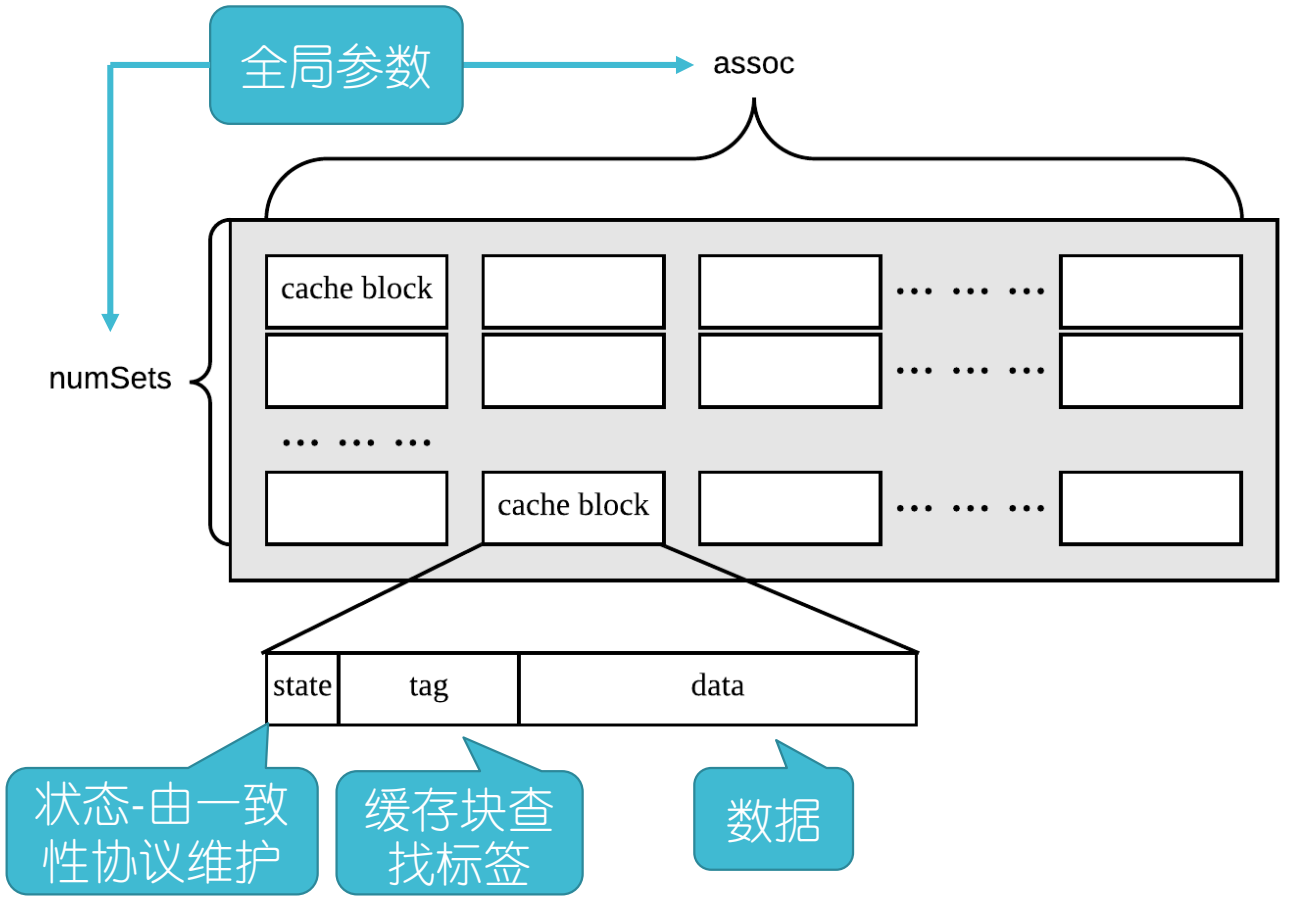


北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

基于深度学习的数据预取质量优化算法设计与实现

存储模块的设计与实现

- Cache blocks模块设计

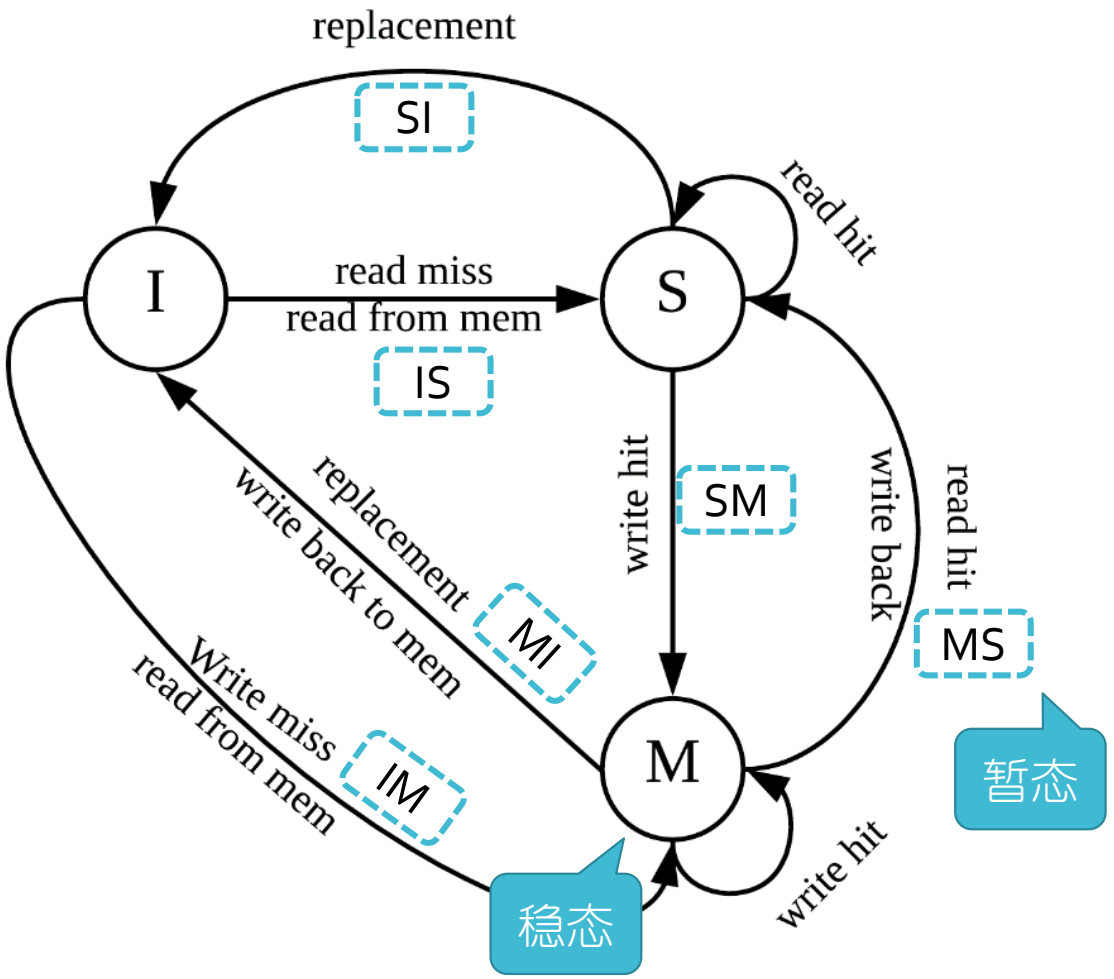


北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

基于深度学习的数据预取质量优化算法设计与实现

存储模块的设计与实现

- MSI一致性协议设计



I: Invalid
S: Shared
M: Modified

北京工业大学 BEIJING UNIVERSITY OF TECHNOLOGY

基于深度学习的数据预取质量优化算法设计与实现

存储模块的设计与实现

LFU替换算法设计

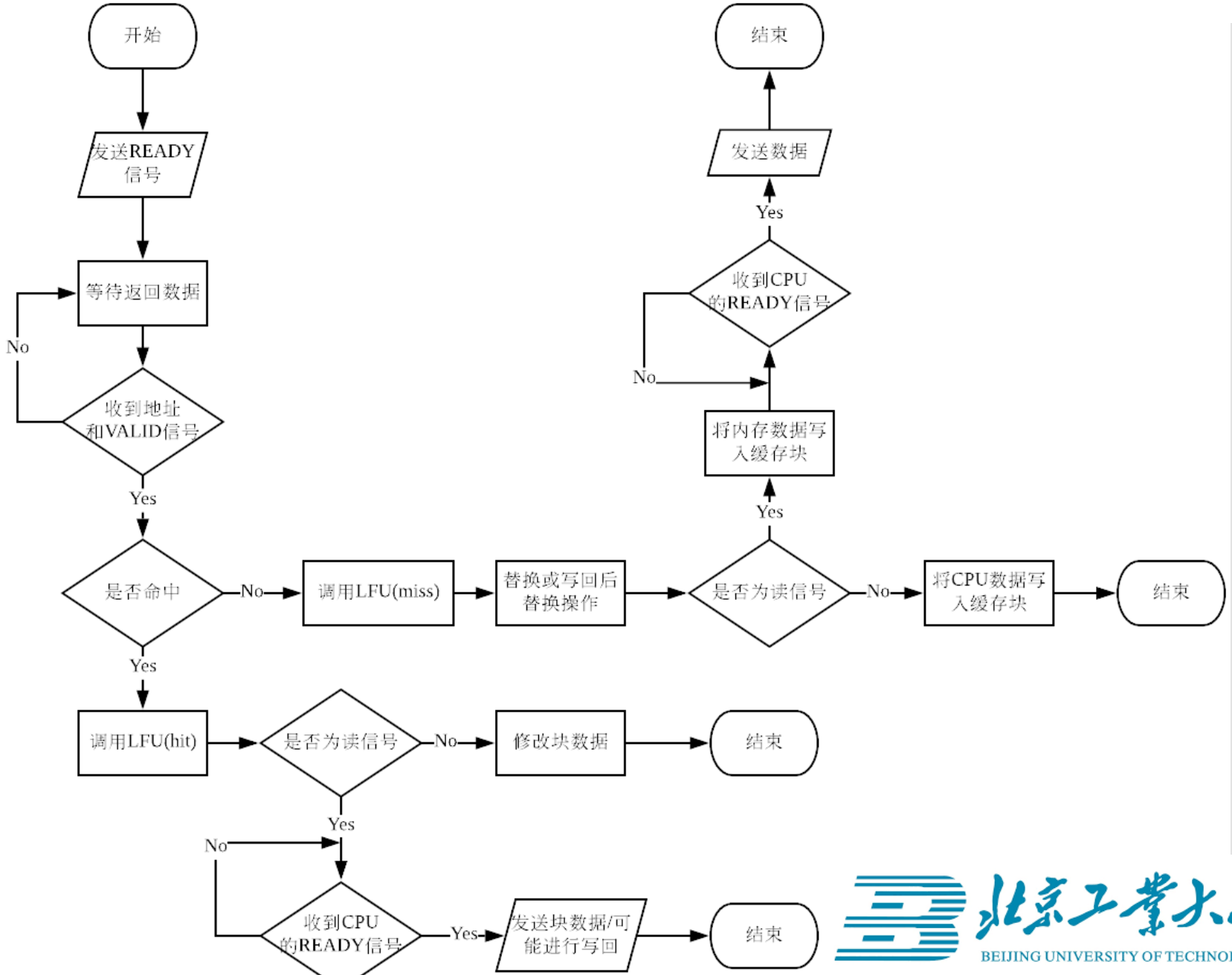
缓存块命中操作		缓存块缺失操作
def hit(hitWay) hitCounterValue = counter(hitWay).value for each counter : if counter.value <= hitCounterValue : counter.value++ else no action counter(hitWay).value = 0		def miss for each counter : if counter.value > victimCounterValue : victimWay = counter.way victimCounterValue = counter.value else no action for each counter : if counter.way == victimWay : counter.value = 0 else counter.value++ return victimWay

北京工業大學 BEIJING UNIVERSITY OF TECHNOLOGY

- Cache对CPU请求的处理过程

基于深度学习的数据预取质量优化算法设计与实现

存储模块的设计与实现

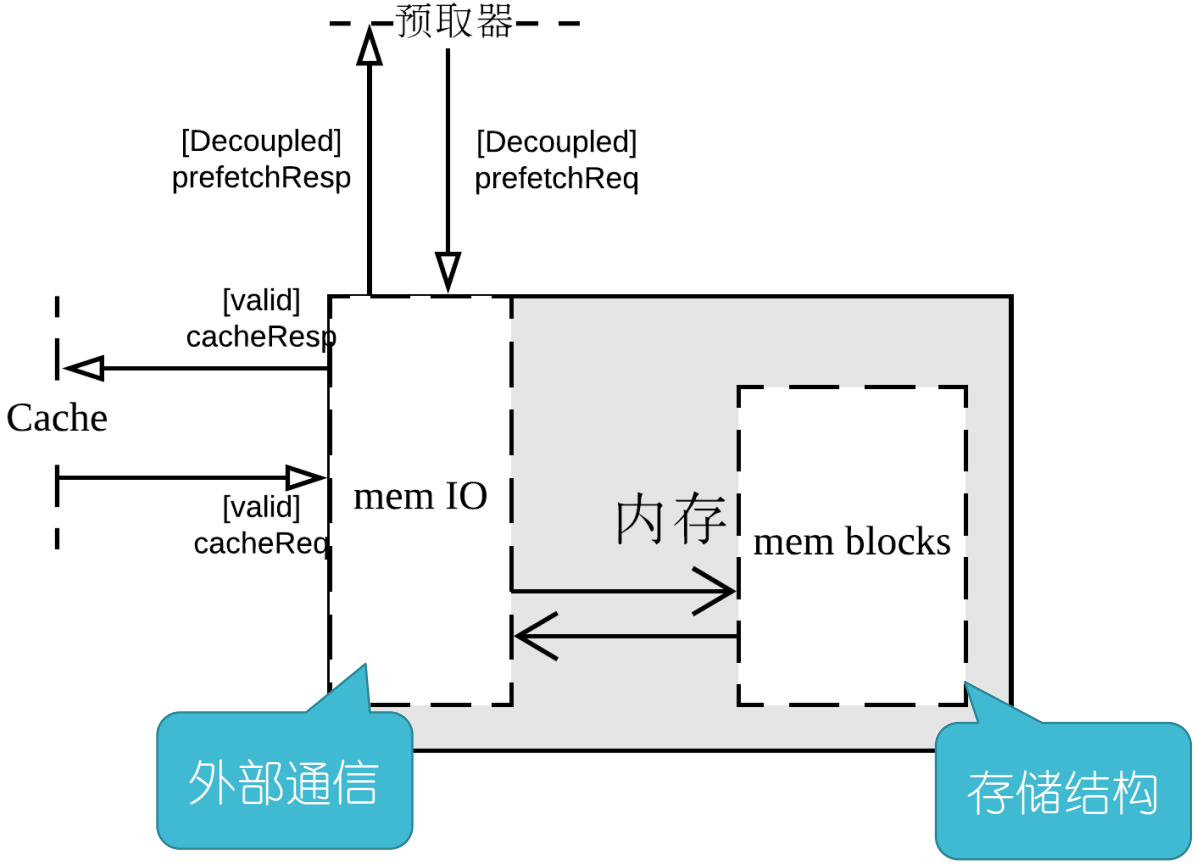


北京工業大學 BEIJING UNIVERSITY OF TECHNOLOGY

基于深度学习的数据预取质量优化算法设计与实现

存储模块的设计与实现

- 内存模块设计



北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

基于深度学习的数据预取质量优化算法设计与实现

预取器的设计与实现

- Next Line Prefetcher

不论cache是否命中，都发出预取命令，将被访问字所在块的下一块从内存预取到cache中。

```
if CPU.request.valid == true :
  response.valid = true
  response. prefetchTarget = CPU.request. effectiveAddress + addrWidth
  send response. prefetchTarget to cache
```

预取器的设计与实现

- Stride Prefetcher
- 基本原理

假设未来将要访问的地址流：a_1、 a_2、 a_3…

满足下列公式时

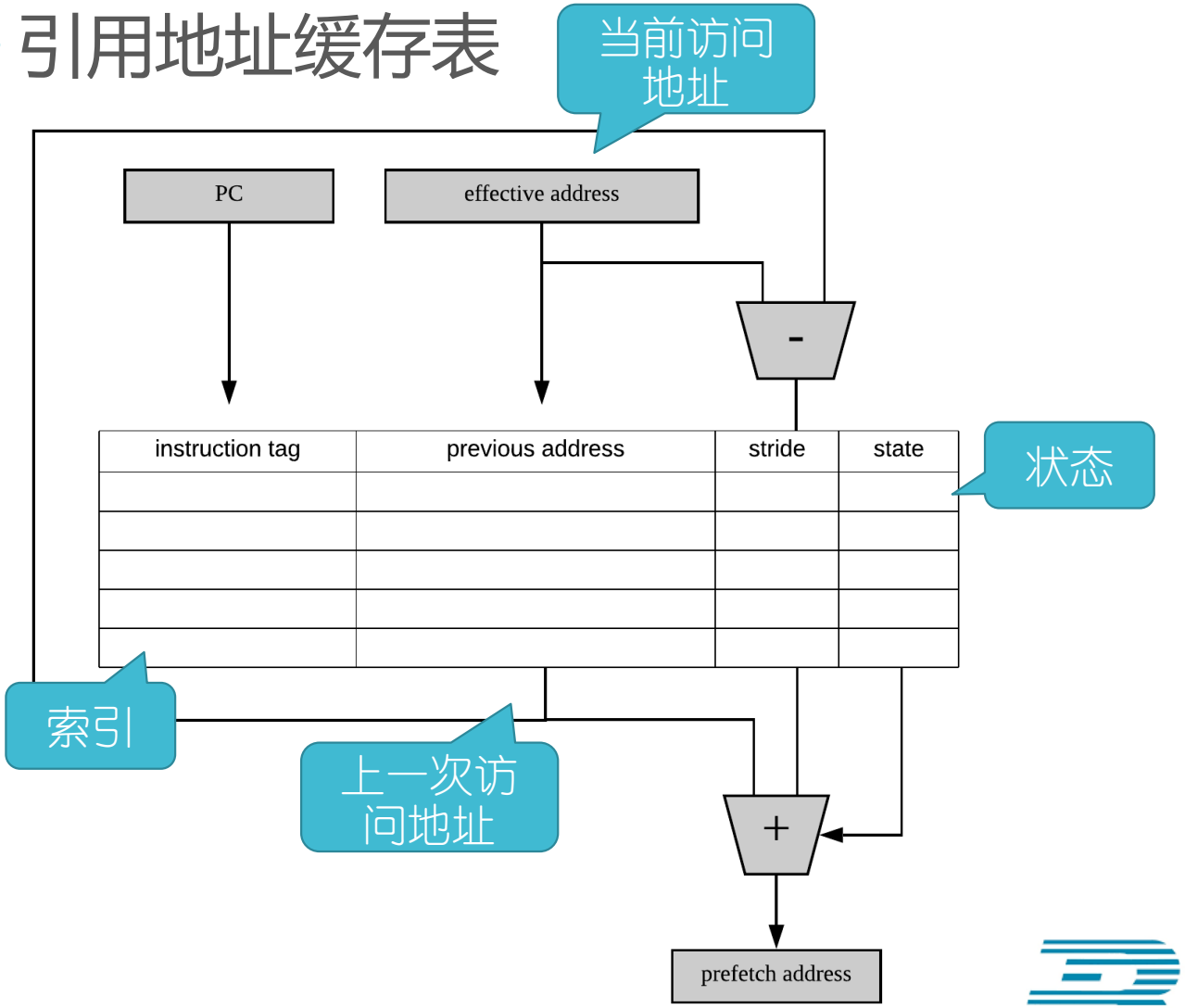
$$(a_2 - a_1) = \Delta \neq 0$$
$$A_3 = a_2 + \Delta$$
$$A_n == a_n$$

发出预取请求

基于深度学习的数据预取质量优化算法设计与实现

预取器的设计与实现

- Stride Prefetcher
- 引用地址缓存表

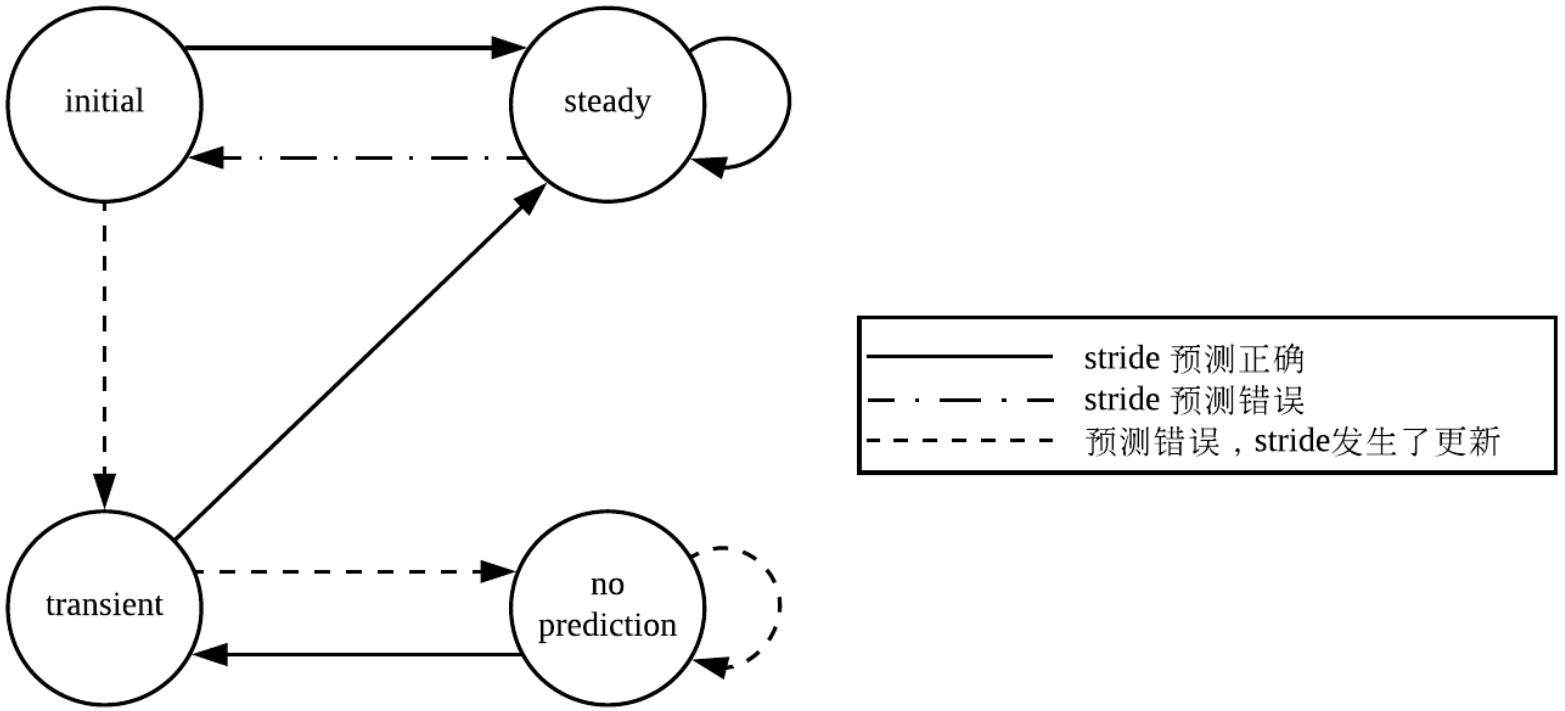


北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

基于深度学习的数据预取质量优化算法设计与实现

预取器的设计与实现

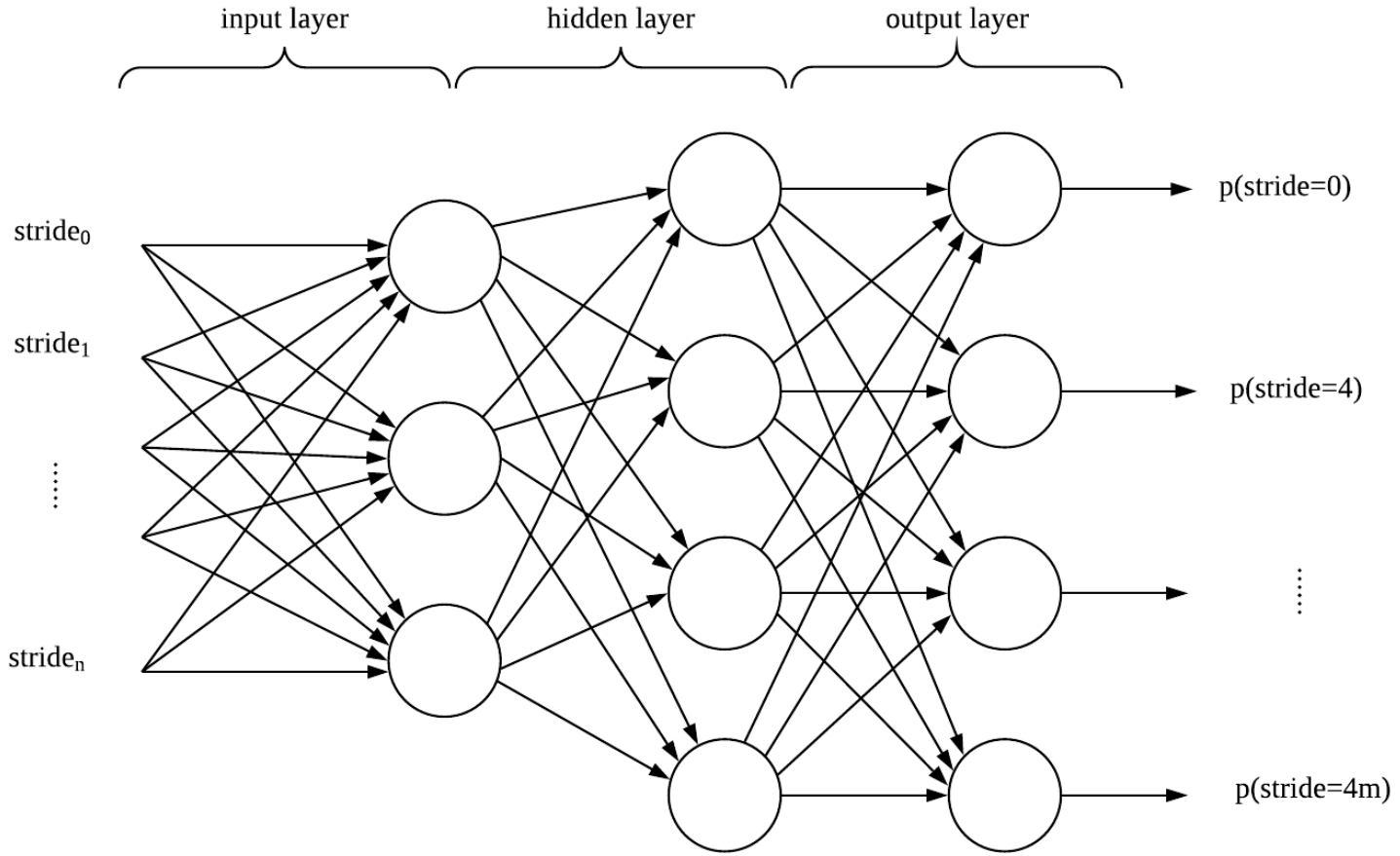
- Stride Prefetcher
- 表项状态值的维护



基于深度学习的数据预取质量优化算法设计与实现

预取器的设计与实现

- 基于感知机的Stride Prefetcher设计



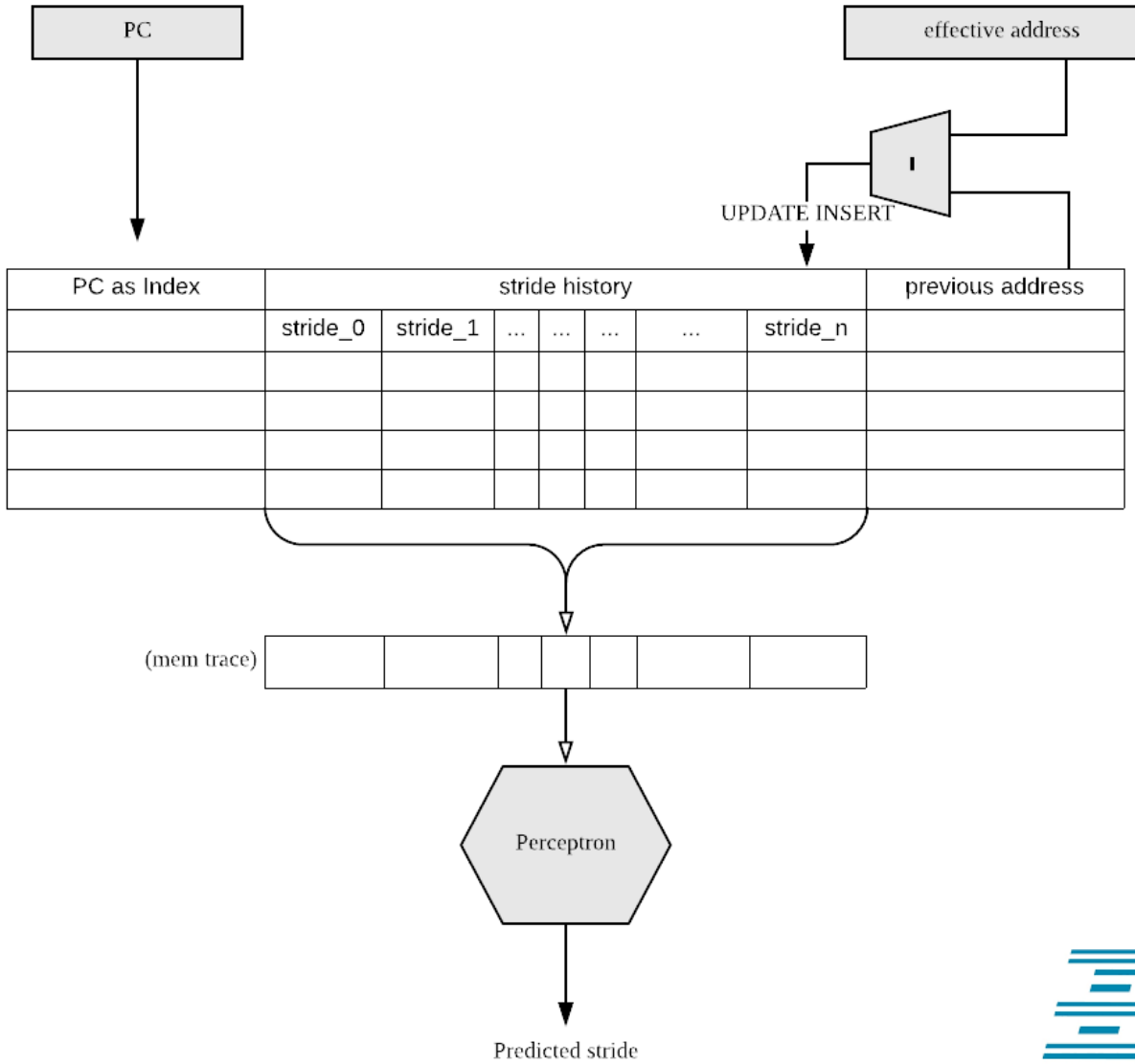
输出$y_1,\ldots,y_m$，则有$\Sigma\{y_1,\ldots,y_m\}=1$

北京工業大學 BEIJING UNIVERSITY OF TECHNOLOGY

基于深度学习的数据预取质量优化算法设计与实现

预取器的设计与实现

- 基于感知机的Stride Prefetcher设计



替代缓存表和状态机

基于深度学习的数据预取质量优化算法设计与实现

总结与改进目标

后续工作：

- 系统集成
- 整体测试
- 多预取器控制优化
- ……

谢谢

北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY