华中科技大学

计算机系统结构——复习课

童　薇

tongwei@hust.edu.cn

4个定量原理

1. 以经常性事件为重点

- 对经常发生的情况采用优化方法的原则进行选择，以得到更多的总体上的改进
- 优化是指分配更多的资源、达到更高的性能或者分配更多的电能等

2. Amdahl定律

加快某部件执行速度所能获得的系统性能加速比，受限于该部件的执行时间占系统中总执行时间的百分比。

系统性能加速比：

$$\text{加速比} = \frac{\text{系统性能}_{\text{改进后}}}{\text{系统性能}_{\text{改进前}}} = \frac{\text{总执行时间}_{\text{改进前}}}{\text{总执行时间}_{\text{改进后}}}$$

3. CPU性能公式

- 执行一个程序所需的CPU时间

CPU时间 = 执行程序所需的时钟周期数×时钟周期时间

其中：时钟周期时间是系统时钟频率的倒数。

- 每条指令执行的平均时钟周期数CPI

（Cycles Per Instruction）

CPI = 执行程序所需的时钟周期数 / IC

IC：所执行的指令条数

- 程序执行的CPU时间可以写成

CPU时间 = IC ×CPI ×时钟周期时间

4. 程序的局部性原理 (Locality)

程序执行时所访问的存储器地址分布不是随机的，而是相对地簇聚。

- 常用的一个经验规则

  程序执行时间的90%都是在执行程序中10%的代码。

- 程序的时间局部性 (Temporal Locality)

  程序即将用到的信息很可能就是目前正在使用的信息。

- 程序的空间局部性 (Spatial Locality)

  程序即将用到的信息很可能与目前正在使用的信息在空间上相邻或者临近

Make the common case faster | Amdahl

$$S_n = \frac{To}{T_n} = \frac{1}{(1 - Fe) + \frac{Fe}{Se}}$$

CPU性能公式

CPU时间 = CPI × IC × 时钟周期时间

CPU时间 = Σ ($CPI_i \times IC_i$) / 时钟频率

CPI = Σ ($CPI_i \times IC_i$) / IC = Σ ($CPI_i \times IC_i$ / IC)

MIPS（Million Instructions Per Second）

$$\text{MIPS} = \frac{\text{指令条数}}{\text{执行时间} \times 10^6} = \frac{\text{指令条数}}{\text{CPU时钟周期数} \times 10^6 / f} = \frac{f}{\text{CPI} \times 10^6}$$

$$\text{程序的执行时间Te} = \frac{\text{指令条数}}{\text{MIPS} \times 10^6}$$

MIPS指令格式

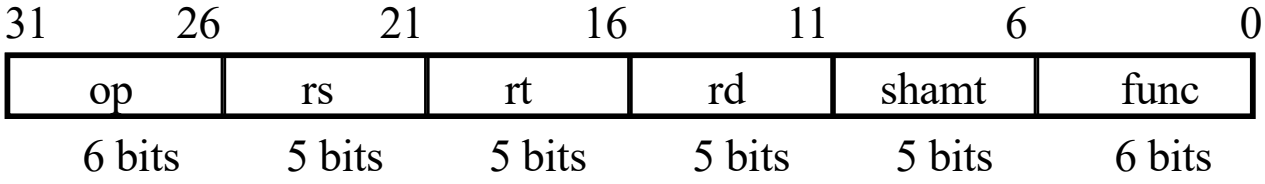
- 所有指令都是32位宽，须按字地址对齐

◆ **有三种指令格式**

– **R-Type**

两个操作数和结果都在寄存器的运算指令。如：sub rd, rs, rt

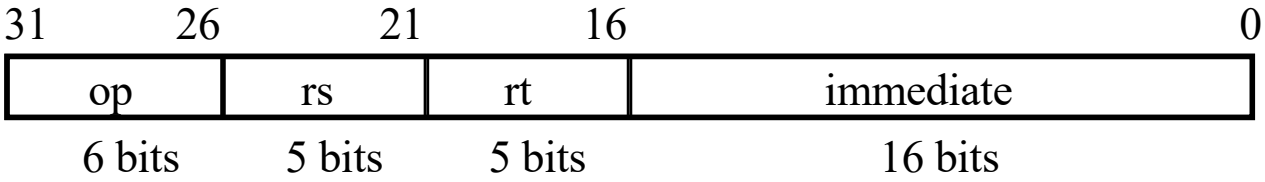
R-Type指令



– **I-Type**

- **运算指令：一个寄存器、一个立即数。如：ori rt, rs, imm16**
- **LOAD和STORE指令。如：lw rt, rs, imm16**
- **条件分支指令。如：beq rs, rt, imm16**

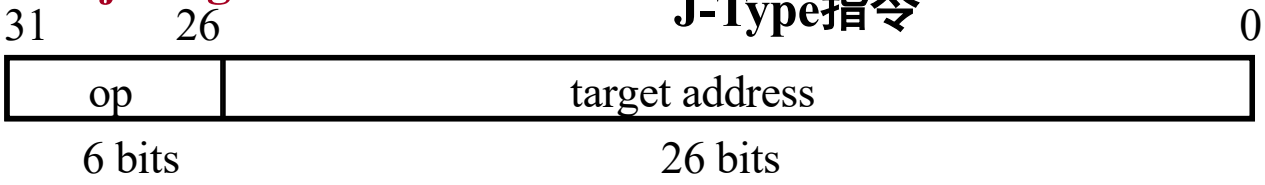
I-Type指令



– **J-Type**

无条件跳转指令。如：j target

J-Type指令



MIPS指令字段含义

OP：操作码

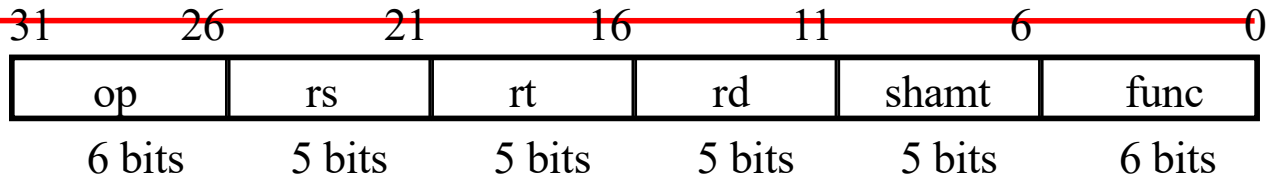
rs：第一个源操作数寄存器

rt：第二个源操作数寄存器

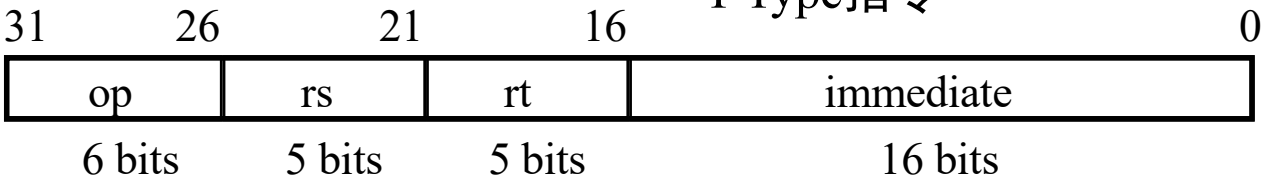
rd：结果寄存器

shamt：移位指令的位移量

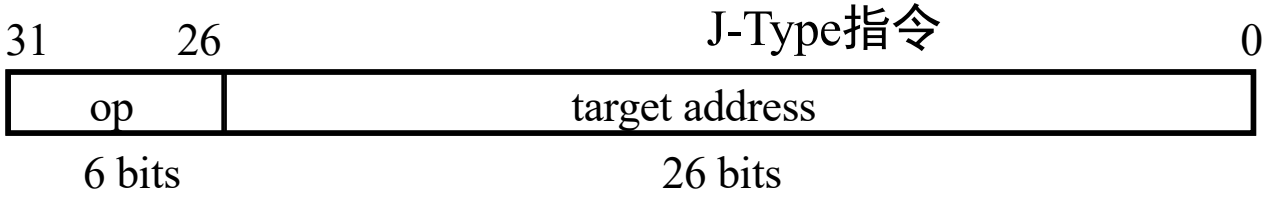
R-Type指令



I-Type指令



J-Type指令



func：R-Type指令的OP字段是特定的“000000”，具体操作由func字段给定。例如：func=“100000”时，表示“加法”运算。

操作码的不同编码定义不同的含义，操作码相同时，再由功能码定义不同的含义！

immediate：立即数或load/store指令和分支指令的偏移地址

target address：无条件转移地址的低26位。将PC高4位拼上26位直接地址，最后添2个“0”就是32位目标地址。为何最后两位要添“0”？

指令按字地址对齐，所以每条指令的地址都是4的倍数（最后两位为0）。

load和store指令

指令举例	指令名称	含 义
LD R2，20(R3)	装入双字	Regs[R2] $\leftarrow_{64}$ Mem[20+Regs[R3]]
LW R2，40(R3)	装入字	Regs[R2] $\leftarrow_{64}$ (Mem[40+Regs[R3]]$_0$)^{32} ## Mem[40+Regs[R3]]
LB R2，30(R3)	装入字节	Regs[R2] $\leftarrow_{64}$ (Mem[30+Regs[R3]]$_0$)^{56} ## Mem[30+Regs[R3]]
LBU R2，40(R3)	装入无符号字节	Regs[R2] $\leftarrow_{64}$ 0^{56} ## Mem[40+Regs[R3]]
LH R2，30(R3)	装入半字	Regs[R2] $\leftarrow_{64}$ (Mem[30+Regs[R3]]$_0$)^{48} ## Mem[30+Regs[R3]]## Mem[31+Regs[R3]]
L.S F2，60(R4)	装入半字	Regs[F2] $\leftarrow_{64}$ Mem[60+Regs[R4]] ## 0^{32}
L.D F2，40(R3)	装入双精度浮点数	Regs[F2] $\leftarrow_{64}$ Mem[40+Regs[R3]]
SD R4，300(R5)	保存双字	Mem[300+Regs[R5]] $\leftarrow_{64}$ Regs[R4]
SW R4，300(R5)	保存字	Mem[300+Regs[R5]] $\leftarrow_{32}$ Regs[R4]
S.S F2，40(R2)	保存单精度浮点数	Mem[40+Regs[R2]] $\leftarrow_{32}$ Regs[F2] $_{0\cdots31}$
SH R5，502(R4)	保存半字	Mem[502+Regs[R4]] $\leftarrow_{16}$ Regs[R5] $_{48\cdots63}$

ALU指令

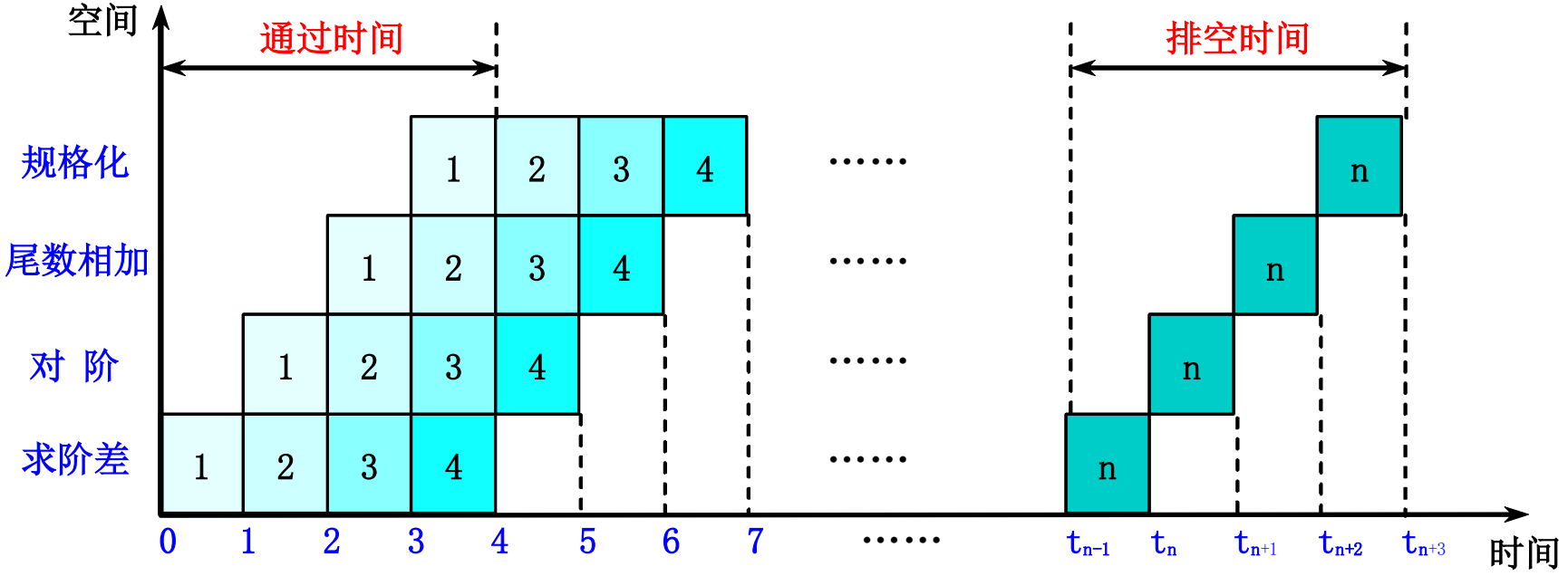
指令举例	指令名称	含义
DADDU R1，R2，R3	无符号加	Regs[R1]← Regs[R2]+ Regs[R3]
DADDIU R4，R5，#6	加无符号立即数	Regs[R4]← Regs[R5]+6
LUI R1，#4	把立即数装入到一个字的高16位	Regs[R1]← 0^{32} ## 4 ## 0^{16}
DSLL R1，R2，#5	逻辑左移	Regs[R1]← Regs[R2]<<5
DSLT R1，R2，R3	置小于	If(Regs[R2]< Regs[R3]) Regs[R1]← 1 else Regs[R1]←0

控制指令

指令举例	指令名称	含义
J name	跳转	PC $_{36..63}$ ← name<<2
JAL name	跳转并链接	Regs[R31] ←PC+4; PC $_{36..63}$ ←name<<2; ((PC+4) −2^{27}) ≤name < ((PC+4) +2^{27})
JALR R3	寄存器跳转并链接	Regs[R31] ←PC+4; PC← Regs[R3]
JR R5	寄存器跳转	PC← Regs[R5]
BEQZ R4, name	等于零时分支	if(Regs[R4]== 0) PC←name ; ((PC+4) −2^{17}) ≤name < ((PC+4) +2^{17})
BNE R3, R4, name	不相等时分支	if(Regs[R3]!= Regs[R4]) PC←name ((PC+4) −2^{17}) ≤name < ((PC+4) +2^{17})
MOVZ R1, R2, R3	等于零时移动	if(Regs[R3]==0) Regs[R1] ← Regs[R2]

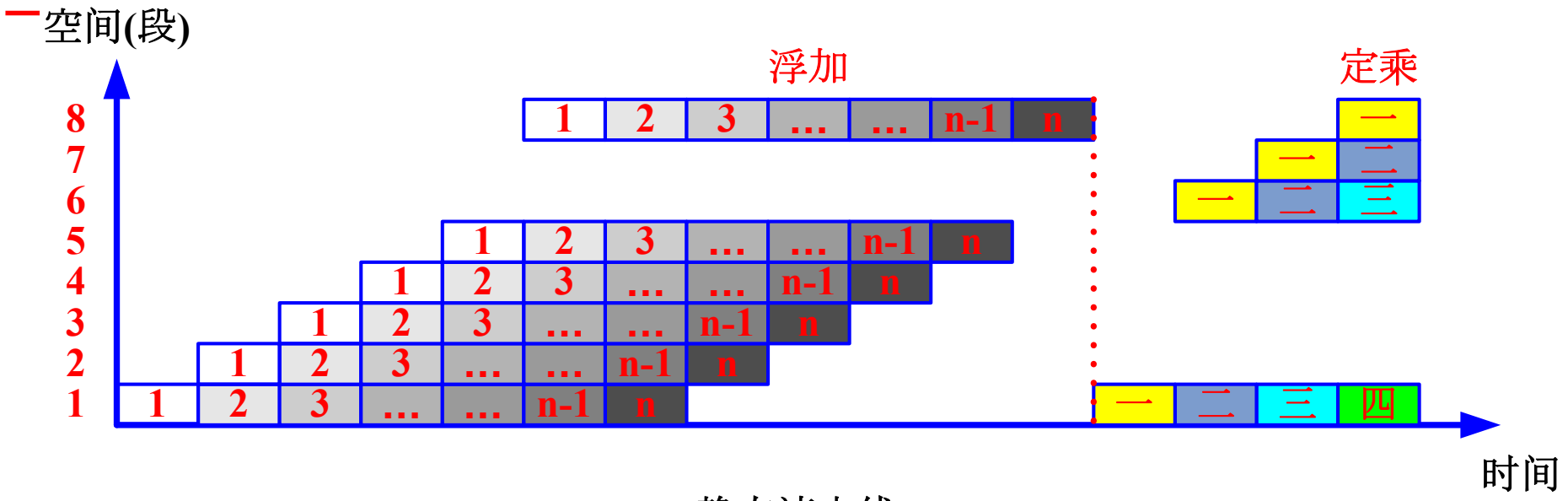
流水线时空图

- 时一空图从时间和空间两个方面描述了流水线的工作过程。时一空图中，横坐标代表时间，纵坐标代表流水线的各个段。
- 浮点加法流水线的时空图

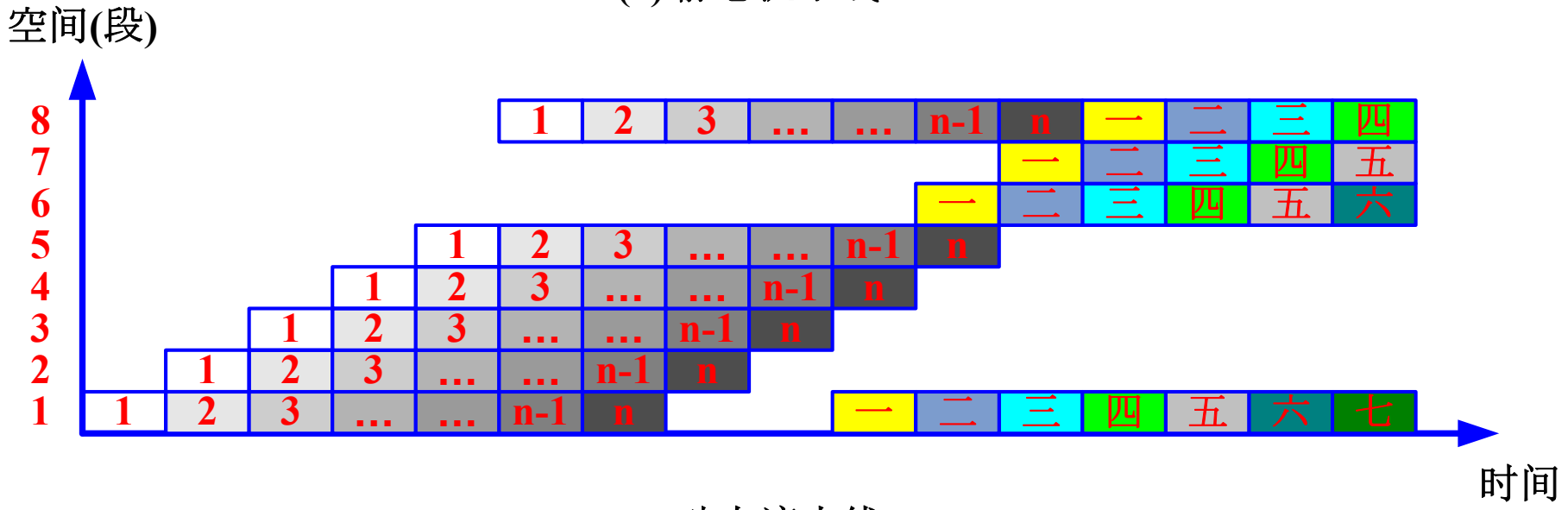


静、动态流水线的时空图

假设该流水线要先做几个浮点加法，然后再做一批定点乘法。



(a) 静态流水线



(b) 动态流水线

流水线的性能指标

吞吐率：在单位时间内流水线所完成的任务数量或输出结果的数量。

$$TP = \frac{n}{T_k}$$

n：任务数

T_k：处理完成n个任务所用的时间

加速比：完成同样一批任务，不使用流水线所用的时间与使用流水线所用的时间之比。

假设：不使用流水线（即顺序执行）所用的时间为T_s，使用流水线后所用的时间为T_k，则该流水线的加速比为：

$$S = \frac{T_s}{T_k}$$

流水线的性能指标

流水线的效率：流水线中的设备实际使用时间与整个运行时间的比值，即流水线设备的利用率。

由于流水线有通过时间和排空时间，所以在连续完成n个任务的时间内，各段并不是满负荷地工作。

单功能非线性流水线的最优调度

- 向一条非线性流水线的输入端连续输入两个任务之间的时间间隔称为非线性流水线的启动距离。
- 会引起非线性流水线功能段使用冲突的启动距离则称为禁用启动距离。
- 启动距离和禁用启动距离一般都用时钟周期数来表示，是一个正整数。
- 预约表
  - 横向（向右）：时间（一般用时钟周期表示）
  - 纵向（向下）：流水线的段

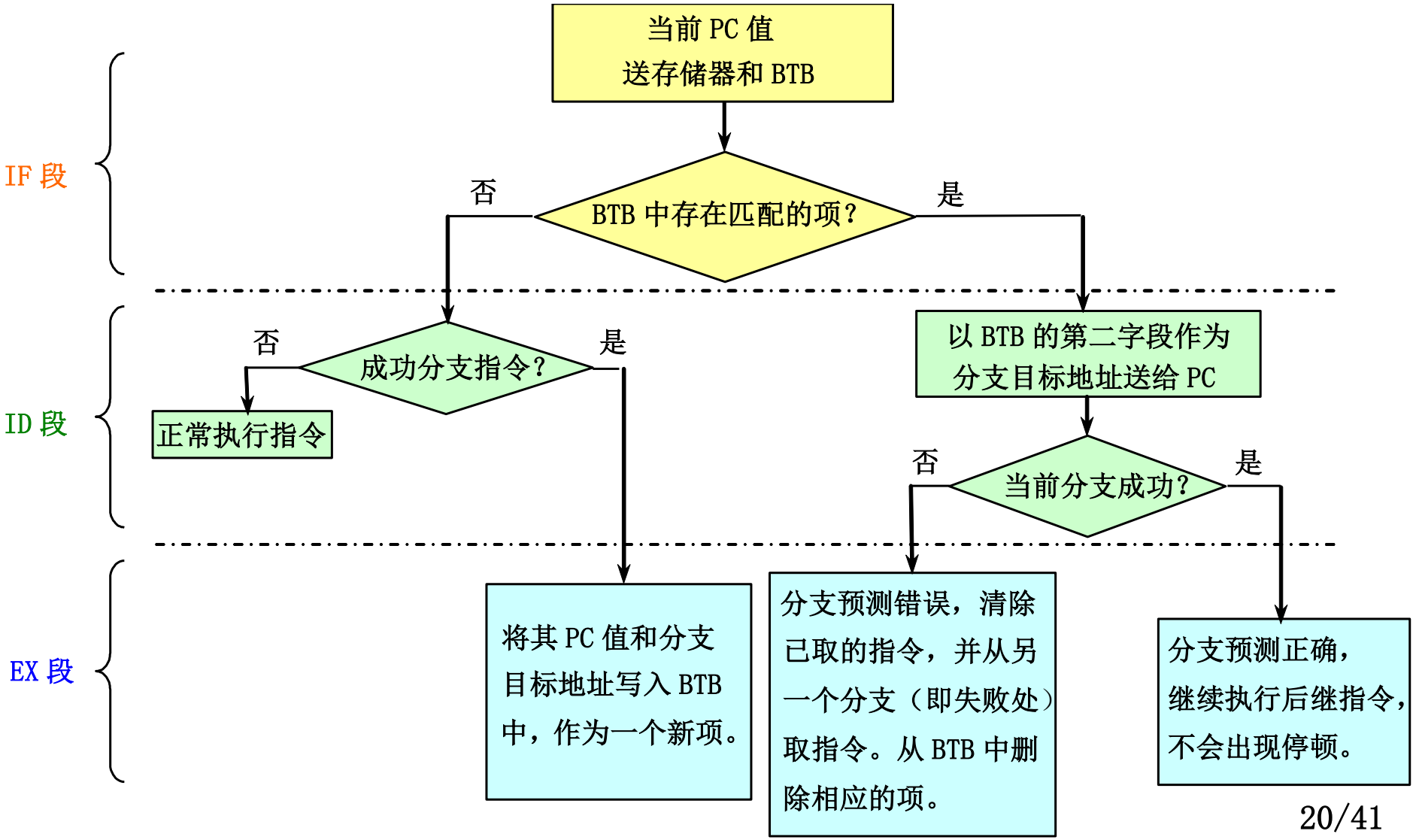
流水线冲突

流水线冲突是指对于具体的流水线来说，由于相关的存在，使得指令流中的下一条指令不能在指定的时钟周期执行。

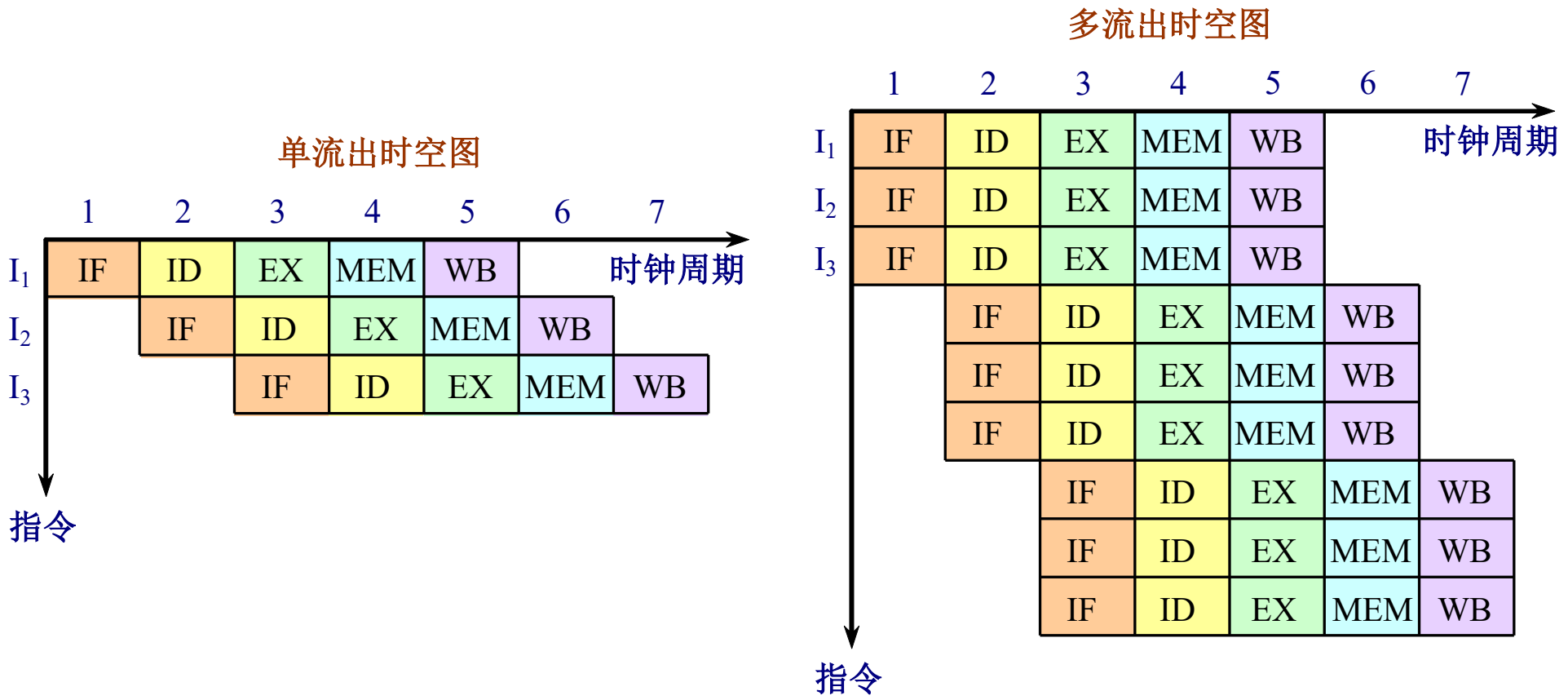
流水线冲突有3种类型：

- 结构冲突：因硬件资源满足不了指令重叠执行的要求而发生的冲突。
- 数据冲突：当指令在流水线中重叠执行时，因需要用到前面指令的执行结果而发生的冲突。
- 控制冲突：流水线遇到分支指令和其它会改变PC值的指令所引起的冲突。

采用分支目标缓冲器BTB

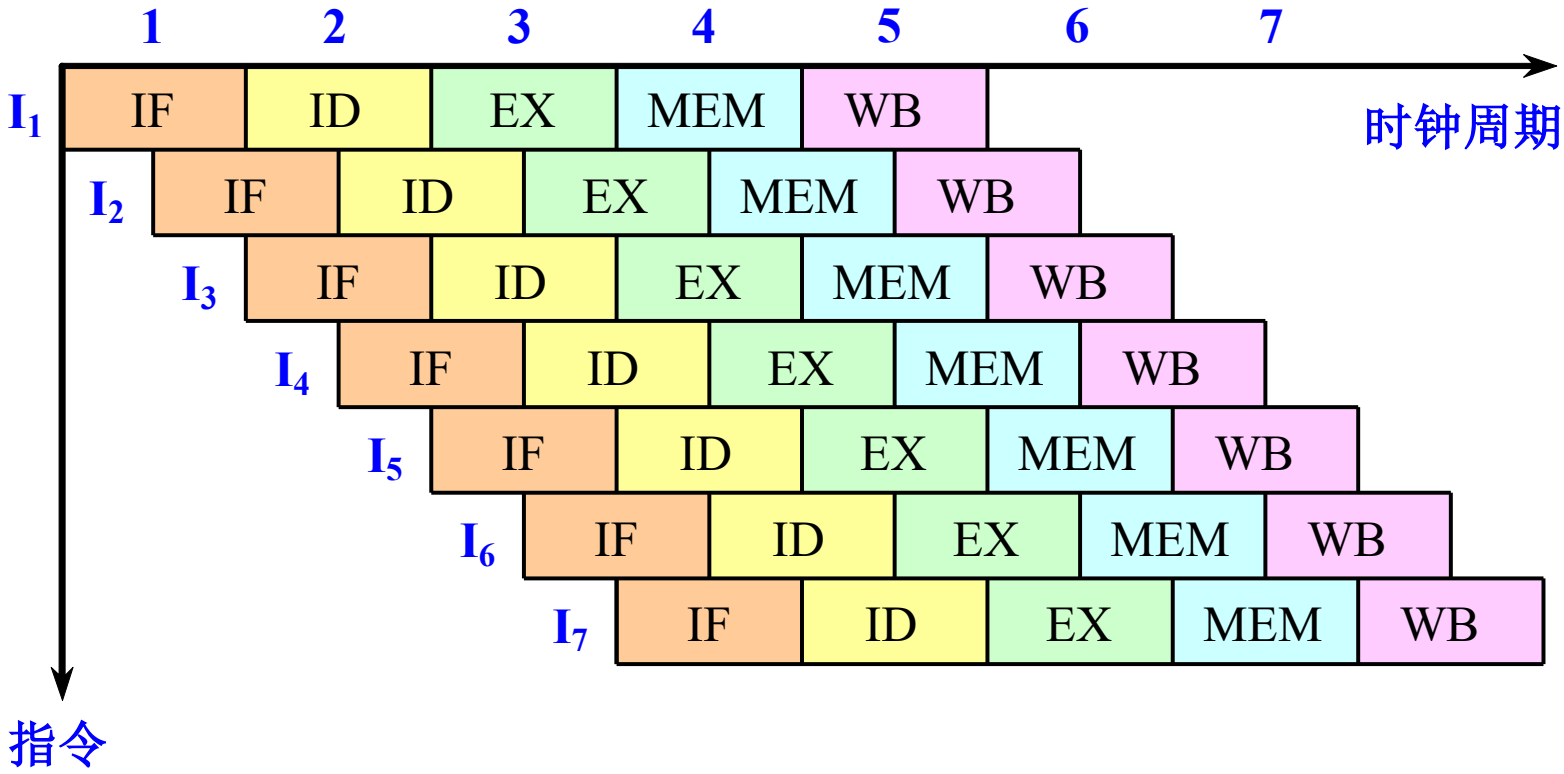


多指令流出技术



单流出和多流出处理机执行指令的时空图

超流水线处理机


1
2
3
4
5
6
7
I1
IF
ID
EX
MEM
WB
时钟周期
I2
IF
ID
EX
MEM
WB
I3
IF
ID
EX
MEM
WB
I4
IF
ID
EX
MEM
WB
I5
IF
ID
EX
MEM
WB
I6
IF
ID
EX
MEM
WB
I7
IF
ID
EX
MEM
WB
指令

1. 存储容量S

   - 一般来说，整个存储系统的容量即是第二级存储器M_2的容量，即$S=S_2$。

2. 每位价格C

$$C=\frac{C_1S_1+C_2S_2}{S_1+S_2}$$

当$S_1 \ll S_2$时，$C \approx C_2$。

3. 命中率*H* 和不命中率*F*

- 命中率：CPU访问存储系统时，在M_1中找到所需信息的概率。

$$H=\frac{N_1}{N_1+N_2}$$

  - N_1 —— 访问M_1的次数
  - N_2 —— 访问M_2的次数

- 不命中率：F＝1－H

存储层次的四个问题

1. 当把一个块调入高一层(靠近CPU)存储器时，可以放在哪些位置上?

   （映象规则）

2. 当所要访问的块在高一层存储器中时，如何找到该块?

   （查找算法）

3. 当发生不命中时，应替换哪一块？

   （替换算法）

4. 当进行写访问时，应进行哪些操作?

   （写策略）

Cache的性能分析

1. 不命中率
   - 与硬件速度无关
   - 容易产生一些误导
2. 平均访存时间

平均访存时间 = 命中时间＋不命中率×不命中开销

3. 程序执行时间

CPU时间＝（CPU执行周期数+存储器停顿周期数）× 时钟周期时间

其中：

- **存储器停顿时钟周期数＝“读”的次数×读不命中率×读不命中开销＋“写”的次数×写不命中率×写不命中开销**
- **存储器停顿时钟周期数＝访存次数×不命中率×不命中开销**

CPU时间＝（CPU执行周期数+访存次数×不命中率×不命中开销）

× 时钟周期时间

$$CPU\text{时间} = IC \times \left(CPI_{execution} + \frac{\text{访存次数}}{\text{指令数}} \times \text{不命中率} \times \text{不命中开销} \right) \times \text{时钟周期时间}$$

$$= IC \times （CPI\text{execution} + \text{每条指令的平均访存次数} \times \text{不命中率} \times \text{不命中开销}） \times \text{时钟周期时间}$$

改进Cache的性能

1. 平均访存时间＝命中时间＋不命中率×不命中开销
2. 可以从三个方面改进Cache的性能：
   - 降低不命中率
   - 减少不命中开销
   - 减少Cache命中时间

快速地址转换技术

1. 地址变换缓冲器TLB

   - TLB是一个专用的高速缓冲器，用于存放近期经常使用的页表项；
   - TLB中的内容是页表部分内容的一个副本；
   - TLB也利用了局部性原理。

2. TLB中的项由两部分构成：标识和数据

   - 标识中存放的是虚地址的一部分。
   - 数据部分中存放的则是物理页帧号、有效位、存储保护信息、使用位、修改位等。

1. 磁盘阵列RAID：使用多个磁盘（包括驱动器）的组合来代替一个大容量的磁盘。
2. RAID 0，1，4，5，10，01的组成及读写访问流程。
3. 可靠度公式

①串并联系统：先串联、后并联，可靠度

$$R(t) = 1-[1-R_i^n(t)]^m$$

②并串联系统：先并联、后串联，可靠度

$$R(t) = [1-(1-R_i(t))^m]^n$$

I/O和Cache数据一致性

1. Cache会使一个数据出现两个副本：

   一个在Cache中，另一个在主存中。

2. I/O设备可以修改存储器中的内容

   - 把I/O连接到存储器上

     会出现以下情况：

     - CPU修改了Cache的内容后，由于存储器的内容跟不上Cache内容的变化，I/O系统进行输出操作时所看到的数据是旧值。（写直达Cache没有这样的问题）
     - I/O系统进行输入操作后，存储器的内容发生了变化，但CPU在Cache中所看到的内容依然是旧值。

4. 平均访问时间T_A

$$T_A = HT_1 + (1-H)(T_1 + T_M)$$
$$= T_1 + (1-H)T_M$$
$$\text{或 } T_A = T_1 + FT_M$$

分两种情况来考虑CPU的一次访存：

- 当命中时，访问时间即为T_1（命中时间）
- 当不命中时，情况比较复杂。

  不命中时的访问时间为：$T_2 + T_B + T_1 = T_1 + T_M$

  $$T_M = T_2 + T_B$$

  - 不命中开销T_M：从向M_2发出访问请求到把整个数据块调入M_1中所需的时间。
  - 传送一个信息块所需的时间为T_B。

互连网络的结构参数与性能指标

互连网络的主要特性参数有：

- 网络规模N：网络中结点的个数。

  表示该网络所能连接的部件的数量。

- 结点度d：与结点相连接的边数（通道数），包括入度和出度。

- 结点距离：对于网络中的任意两个结点，从一个结点出发到另一个结点终止所需要跨越的边数的最小值。

- 网络直径D：网络中任意两个结点之间距离的最大值。

互连网络的结构参数与性能指标

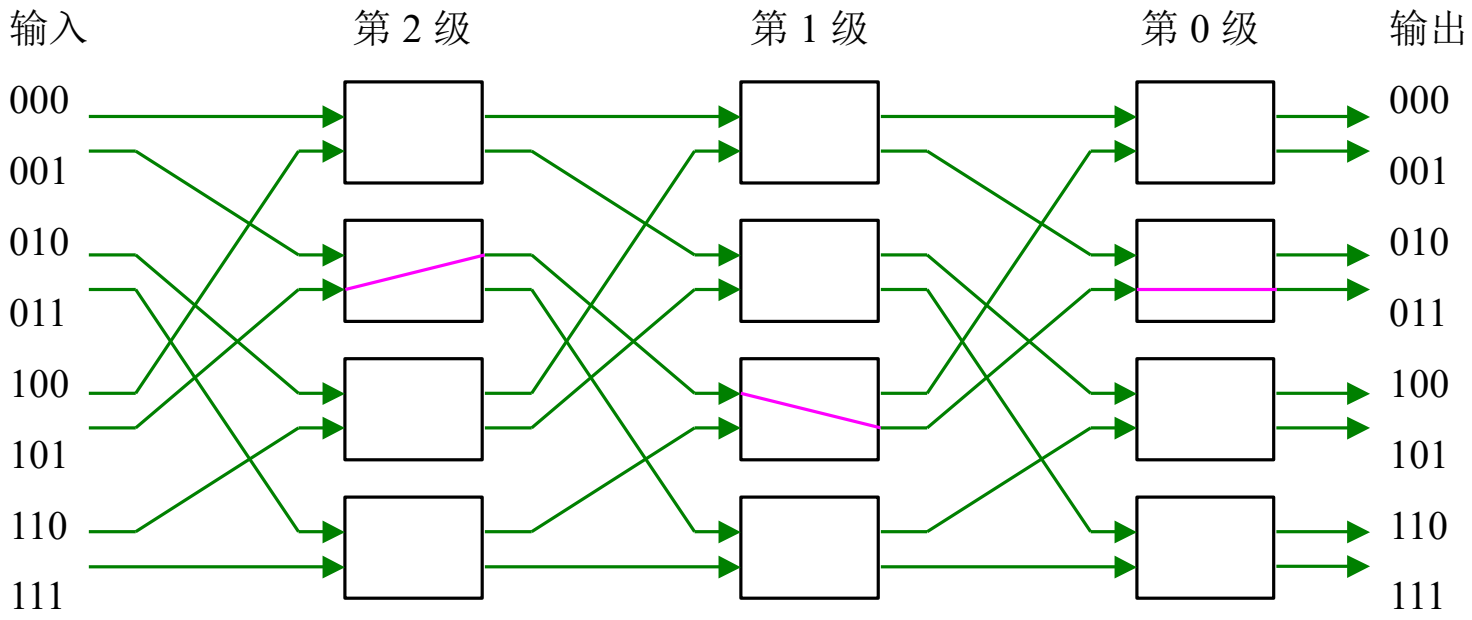
- 等分宽度b：把由N个结点构成的网络切成结点数相同（N/2）的两半，在各种切法中，沿切口边数的最小值。
- 对称性：从任何结点看到的拓扑结构都是相同的网络称为对称网络。

目的：根据给定的输入/输出对应关系，确定各开关的状态。

名称：异或法

操作：将任一个输入端号与它要到达的输出端号作异或运算，其结果的bit_i位控制数据到达的第i级开关，“0”表示“直连”，“1”表示“交换”。

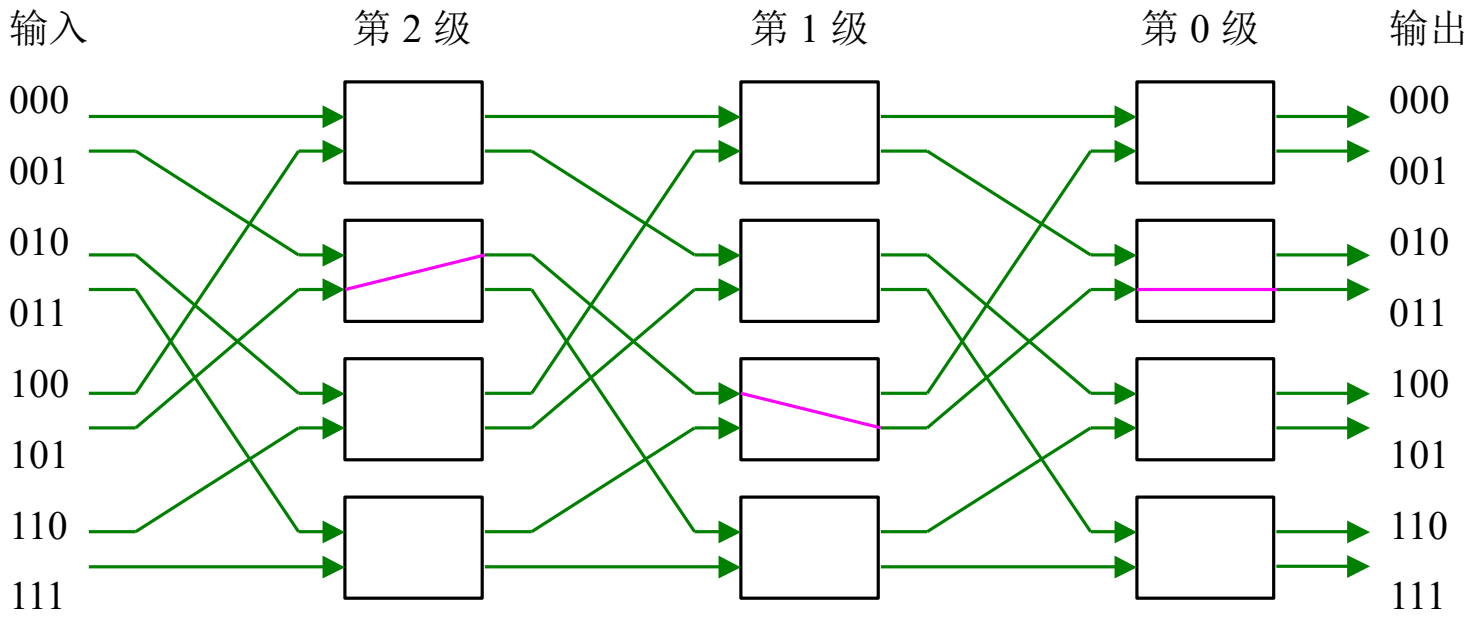
例如给定传输101B→011B，二者异或结果为110B，于是从101B号输入端开始，把它遇到的第2级开关置为“交换”，第1级开关置为“交换”，第0级开关置为“直连”。如下图红线所示。（可简记为“第i级判第i位”）



名称：末址法

操作：输出端号二进制形式的bit_i位控制数据到达的第i级开关，“0”表示从该级开关的上出口输出，“1”表示从该级开关的下出口输出。

例如给定传输101B→011B，输出端号的二进制形式依次为0→1→1，于是从101B号输入端开始，依次选择出口为“上”→“下”→“下”。如下图红线所示。（同样简记为“第i级判第i位”）



并行处理面临的挑战

并行处理面临着两个重要的挑战

- 程序中的并行性有限
- 远程访问延迟

$$\textbf{系统加速比} = \frac{1}{(1-\text{可加速部分比例}) + \dfrac{\text{可加速部分比例}}{\text{理论加速比}}}$$

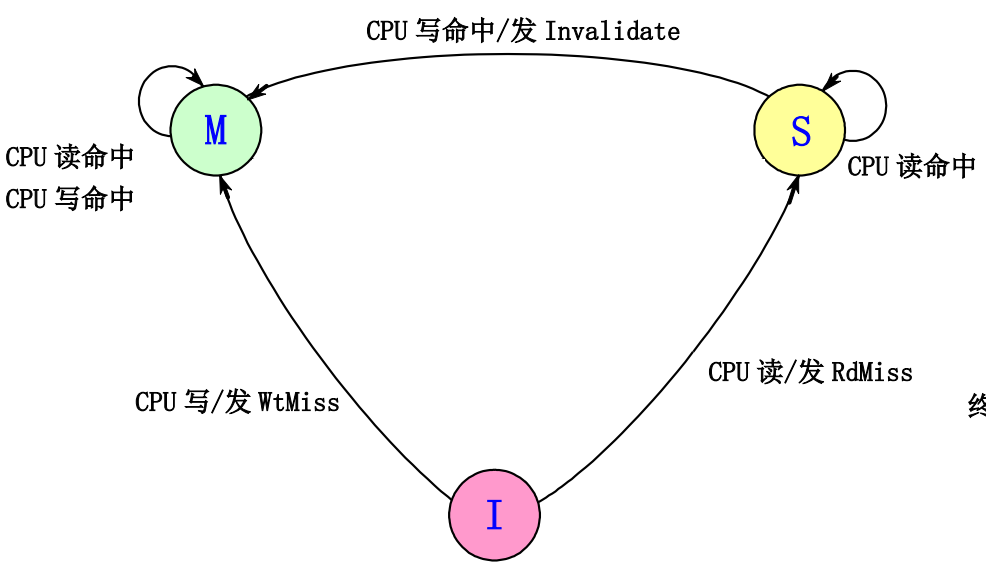
Cache一致性协议

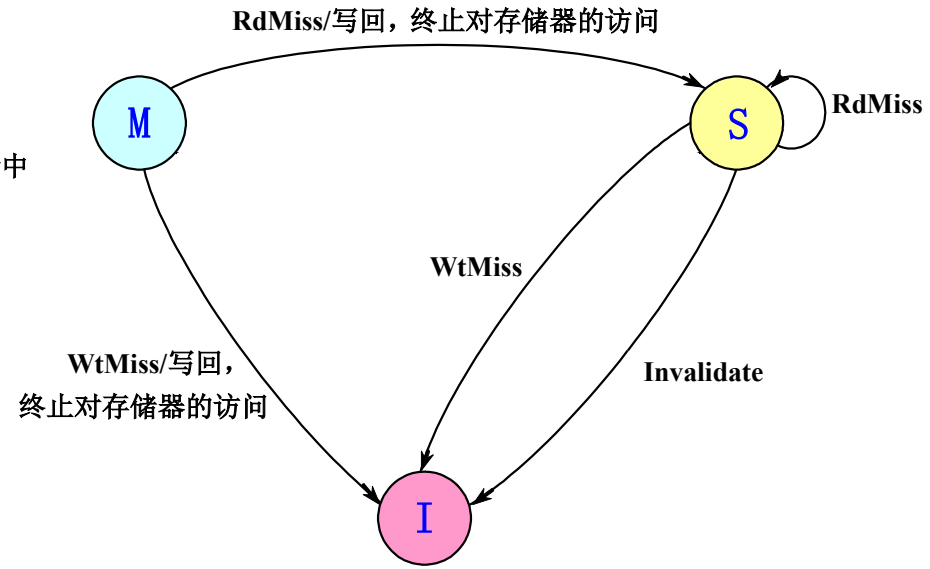
在多个处理器中用来维护一致性的协议。

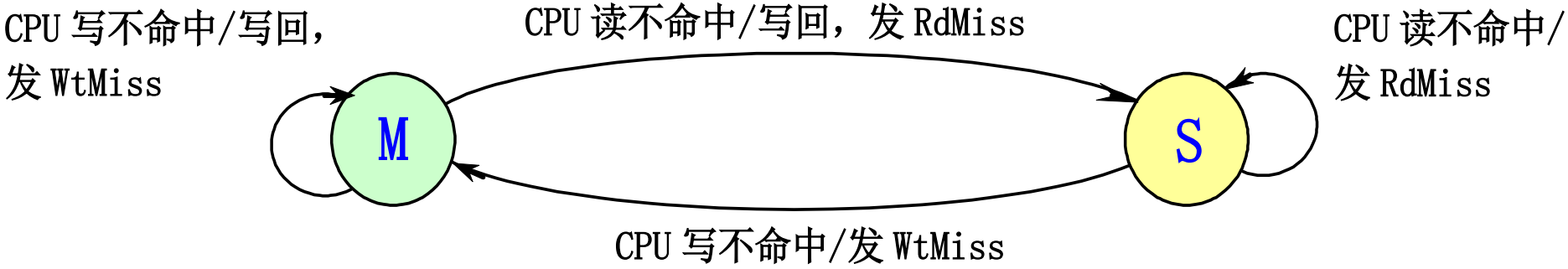
- 关键：跟踪记录共享数据块的**状态**
- 两类协议（采用不同的技术跟踪共享数据的状态）
  - 目录式协议（directory）

    物理存储器中数据块的共享状态被保存在一个称为目录的地方。（一个共享数据块的状态只在一个地方有）
  - 监听式协议（snooping）
    - 每个Cache除了包含物理存储器中块的数据拷贝之外，也保存着各个块的共享状态信息。（每个Cache中都有
- 写作废和写更新

Cache块状态转换图







知识点归纳

第1章：Amdahl定律，CPU时间/CPI/MIPS
第2章：RISC技术，MIPS指令集，MIPS模拟器
第3章：时空图，流水线分类，性能指标，单功能非线性流水线调度，相关，定向，MIPS流水线
第5章：超标量/超长指令字/超流水，分支预测技术
第7章：存储系统性能指标，全相联/直接相联/组相联，查找方法，写策略，不命中类型，伪相联，Cache性能改进技术，两级Cache，虚拟存储器，TLB
第8章：RAID系统及其可靠度公式， I/O与Cache数据一致性
第9章：性能指标，单级互连网络，多级互连网络
第10章：并行处理面临的挑战

预祝各位取得好成绩！