学习内容

- 5.1 重叠方式
- 5.2 流水方式
- 5.3 向量的流水处理与向量处理机
- 5.4 指令级高度并行的超级处理机
- 5.5 ARM流水线处理器举例

5.4 指令级高度并行的超级处理机

- 自20世纪80年代兴起RISC之后，又出现了提高指令级并行（ILP）的新一代处理机，让单处理机在每个时钟周期内解释多条指令。由代表性的例子是：
  - 超标量（Superscalar）处理机
  - 超流水线（Superpipelining）处理机
  - 超长指令字（VLIW）处理机

5.4 指令级高度并行的超级处理机

- **超标量处理机：**
  - Intel公司的i860, i960,
  - Pentium处理机
  - Motolora公司的MC88110
  - IBM公司的Power 6000
  - SUN公司的SuperSPARC等。
- **超流水线处理机：**
  - SGI公司的MIPS R4000, R5000, R10000等。
- **超标量超流水线处理机：**
  - DEC公司的Alpha等。

5.4 指令级高度并行的超级处理机

机器类型	*k*段流水线基准标量处理机	*m*度超标量	*n*度超流水线	(*m*,*n*)度超标量超流水
机器流水线周期	1个时钟周期	1	1/*n*	1/*n*
同时发射指令条数	1条	*m*	1	*m*
指令发射等待时间	1个时钟周期	1	1/*n*	1/*n*
指令级并行度ILP	1	*m*	*n*	*m*×*n*

超标量、超流水、超标量超流水处理机的主要性能

5.4 指令级高度并行的超级处理机

- 指令级并行性(ILP)：指令序列中的并行性
- 思想：可同时流出多个指令/操作
- 表示：

  ILP(m, n)＝m*n,

  其中，

  m— 每个时钟启动的次数；

  n — 每次启动的指令/操作个数；
- 特征：ILP*CPI = 1

5.4.1 超标量处理机

- 1987年提出，其本质就是在不同的流水线中执行不相关指令的能力。
- 常规的标量流水线单处理机是在每个△t时间内解释完一条指令。称这种流水机的度为1。
- 超标量处理机采用 m 条指令流水线（多指令流水线），在每个△t时间内同时解释完 m 条指令。称这种流水机的度为m。

1. 超标量处理机基本结构

- 一般流水线处理机：
  - 一条指令流水线，一个多功能操作部件，每个时钟周期平均执行指令的条数小于1。
- 多操作部件处理机：
  - 一条指令流水线，多个独立的操作部件，操作部件可以采用流水线，也可以不流水。
  - 多操作部件处理机的指令级并行度小于1。

1. 超标量处理机基本结构

- **超标量处理机：**
  - 多条指令流水线。
  - 先进的超标量处理机有：
    - 定点处理部件CPU
    - 浮点处理部件FPU
    - 图形加速部件GPU
    - 大量的通用寄存器
    - 两个一级高速Cache等
  - 超标量处理机的指令级并行度大于1。

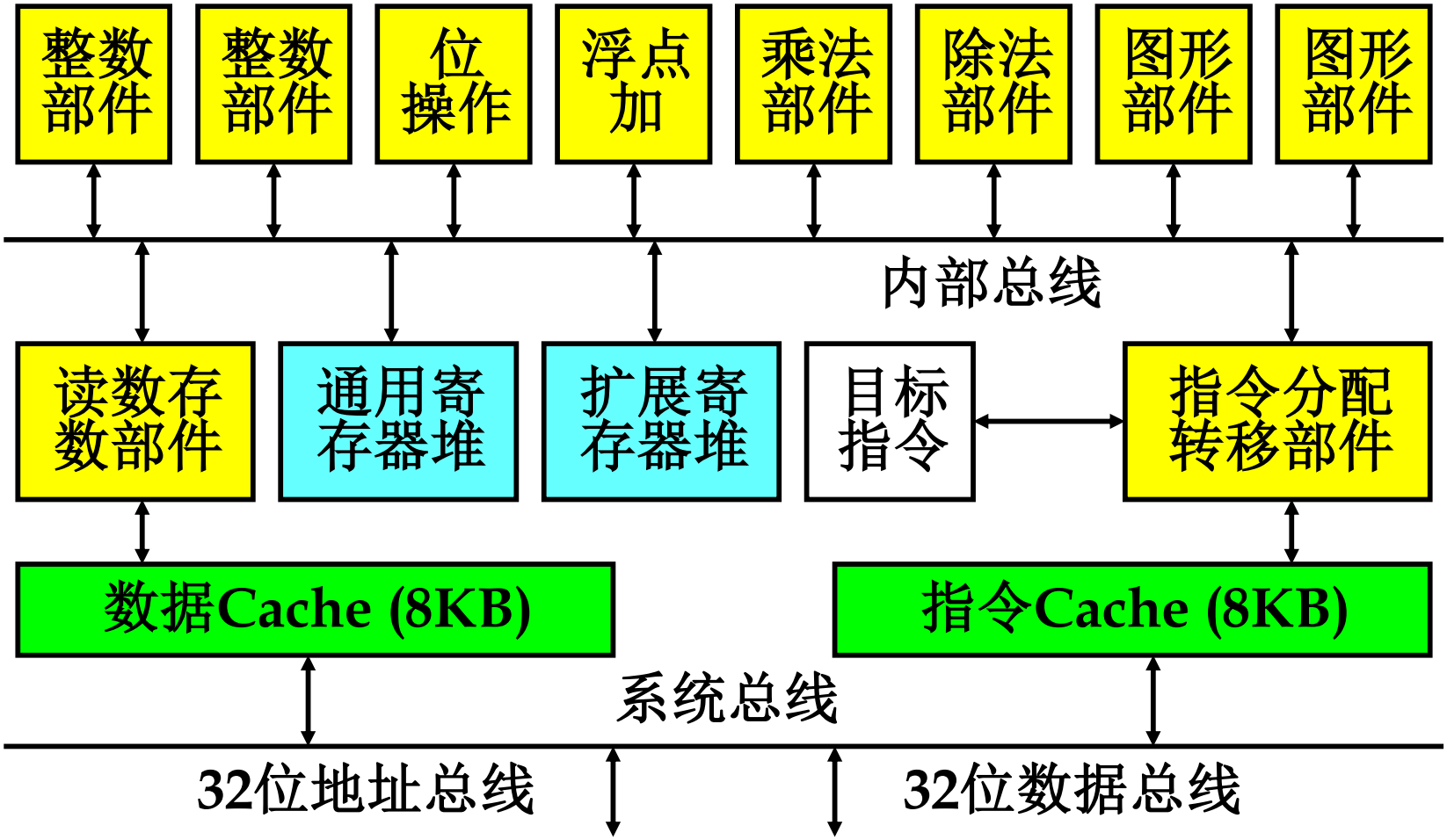
1. 超标量处理机基本结构

- 在超标量处理机中，配置多套功能部件、指令译码电路和多组总线，并且寄存器也备有多个端口和多组总线
- 程序运行时，由指令译码部件检测顺序取出的几条指令之间是否存在数据相关和功能部件争用，将可以并行执行的指令送往流水线，否则就逐条执行

1. 超标量处理机基本结构

- Motorola公司的MC88110：
  - 10个操作部件；
  - 两个寄存器堆：
    - 整数部件通用寄存器堆，32个32位寄存器；
    - 浮点部件扩展寄存器堆，32个80位寄存器；
    - 每个寄存器堆有8个端口，分别与8条内部总线相连接，有一个缓冲深度为4的先行读数栈和一个缓冲深度为3的后行写数栈。
  - 两个独立的高速Cache：
    - 各为8KB，采用两路组相联方式。
  - 转移目标指令Cache：
    - 在有两路分支时，存放其中一路分支上的指令。

1. 超标量处理机基本结构



超标量处理机MC88110的结构

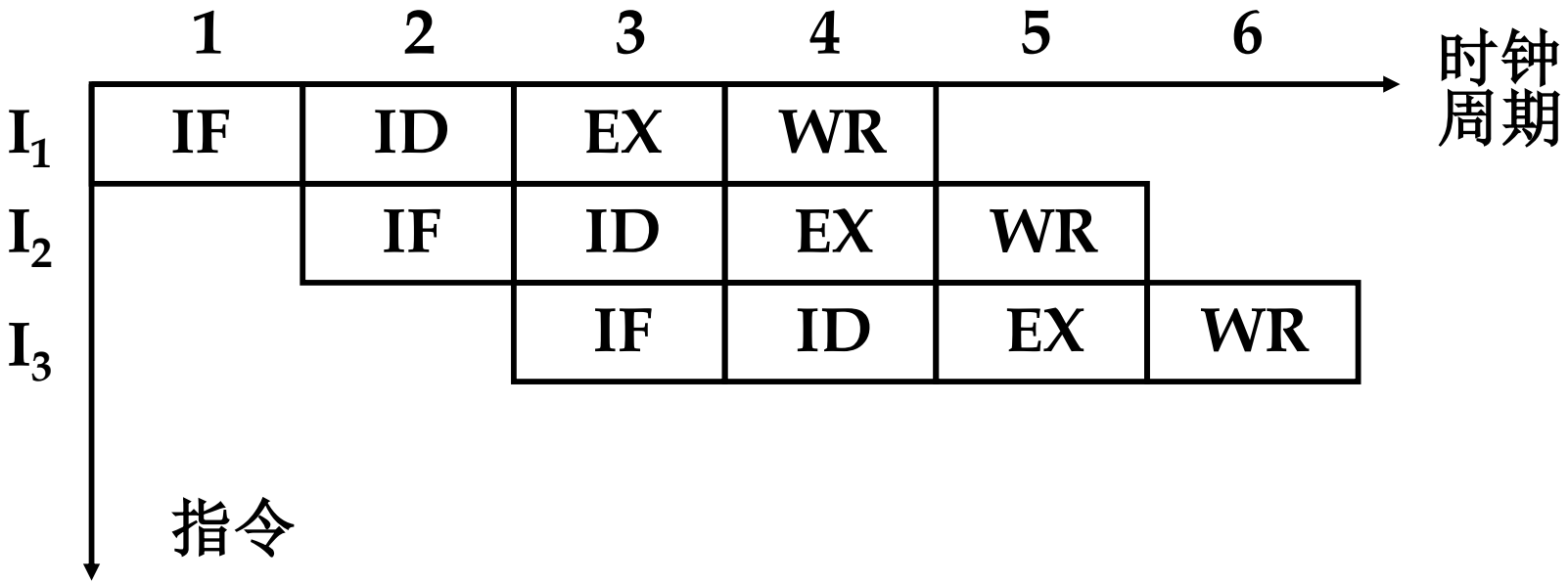
2. 单发射与多发射

- **单发射处理机：**
  - 每个周期只取一条指令、只译码一条指令，只执行一条指令，只写回一个运算结果。
  - 取指部件和译码部件各设置一套。
  - 可以只设置一个多功能操作部件，也可以设置多个独立的操作部件。
  - 操作部件中可以采用流水线结构，也可以不采用流水线结构。
  - 设计目标：每个时钟周期平均执行一条指令，ILP的期望值1。

2. 单发射与多发射

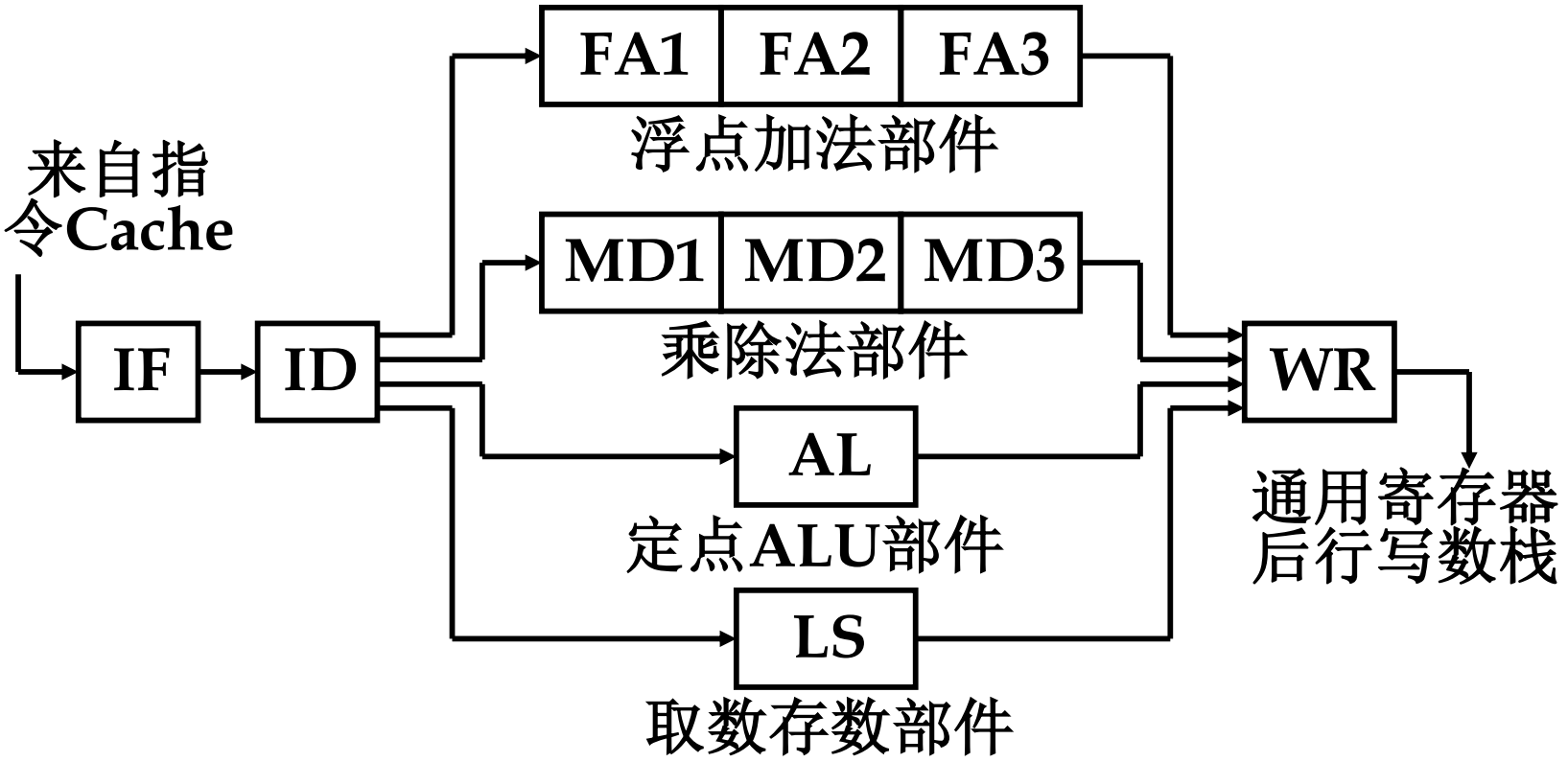
- **单发射处理机：**

单发射处理机的指令流水线时空图



2. 单发射与多发射

- **单发射处理机：**



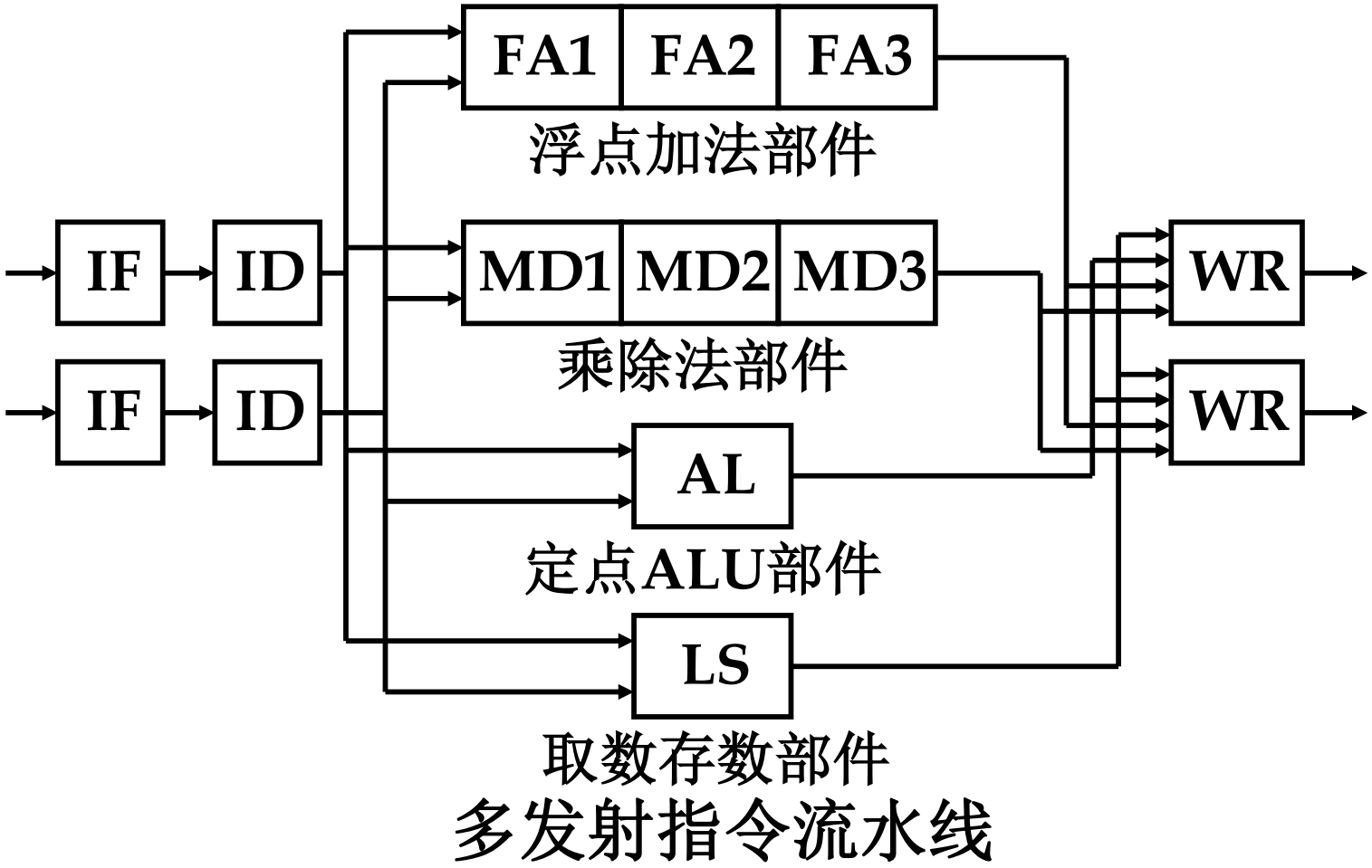
单发射指令流水线

2. 单发射与多发射

- **多发射处理机：**
  - **每个周期同时取多条指令、同时译码多条指令，同时执行多条指令，同时写回多个运算结果。**
  - **需要多个取指令部件，多个指令译码部件和多个写结果部件。**
  - **设置多个指令执行部件，复杂的指令执行部件一般采用流水线结构。**
  - **设计目标：每个时钟周期平均执行多条指令，ILP的期望值大于1。**

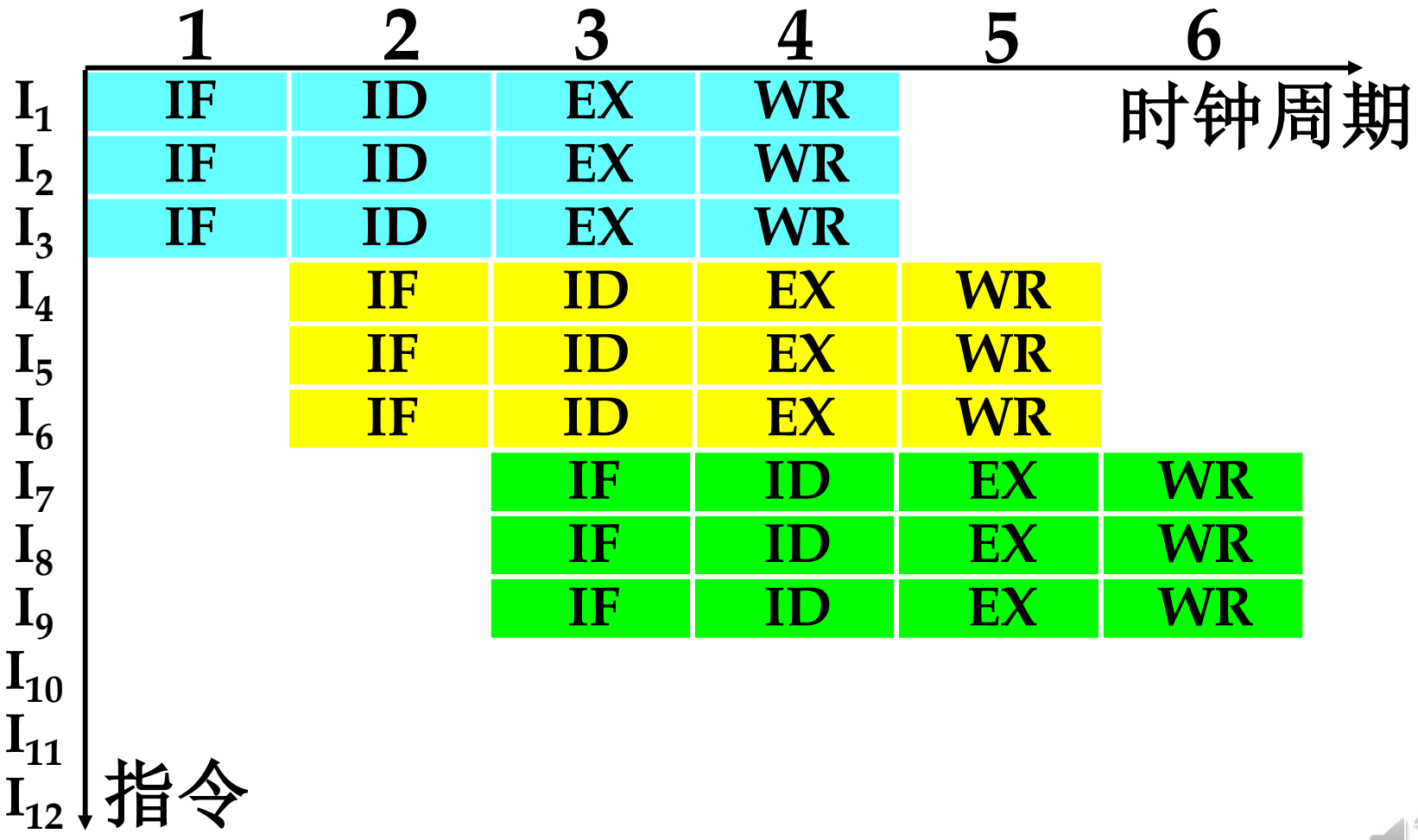
2. 单发射与多发射

- 多发射处理机：



2. 单发射与多发射

3条4段流水线，每时钟周期同时发射3条指令



2. 单发射与多发射

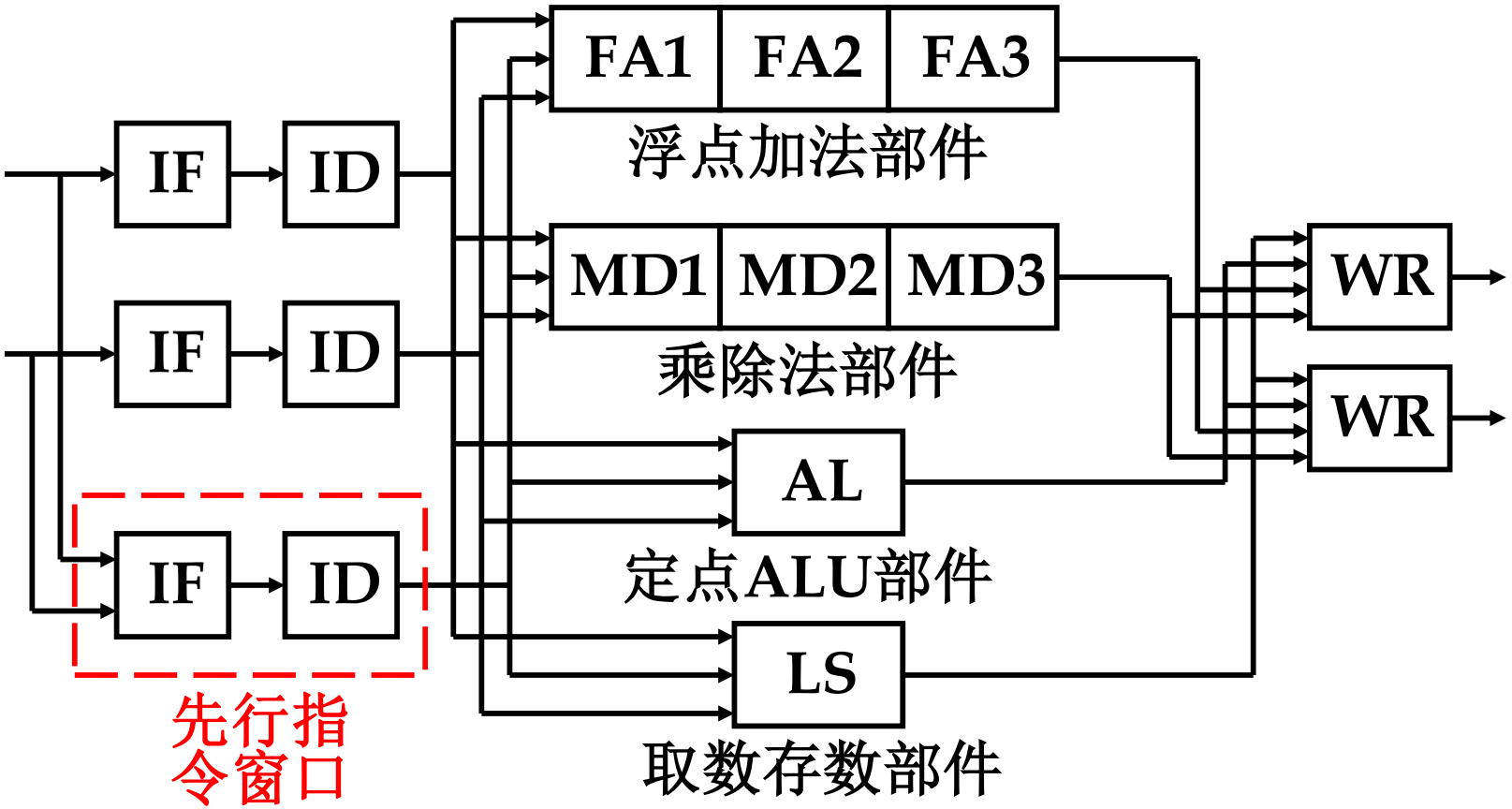
- **超标量处理机：**
  - **一个时钟周期内能够同时发射多条指令的处理机称为超标量处理机。**
  - **必须有两条或两条以上能够同时工作的指令流水线。**
  - **指令级并行度：1 < ILP < m。**

    m 为每个周期同时发射的指令条数。

2. 单发射与多发射

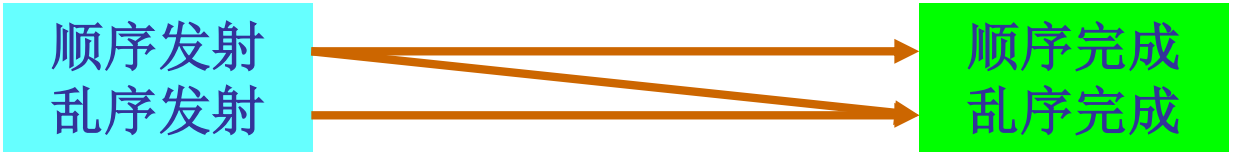
- 先行指令窗口：
  - 能够从指令Cache中预取多条指令。
  - 能够对窗口内的指令进行数据相关性分析和功能部件冲突的检测。
    - 窗口的大小：一般为2至8条指令。
  - 采用目前的指令调度技术，每个周期发射2至4条指令比较合理。

2. 单发射与多发射



有先行指令窗口的多发射指令流水线

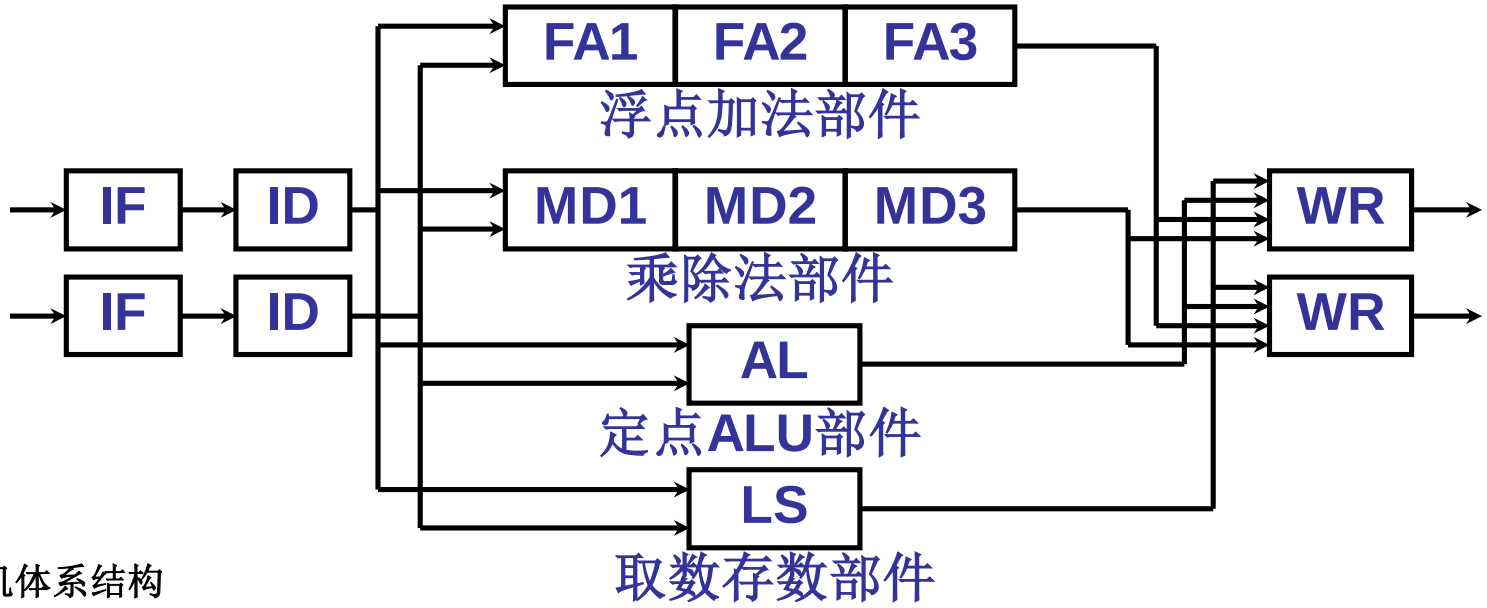
多流水线调度



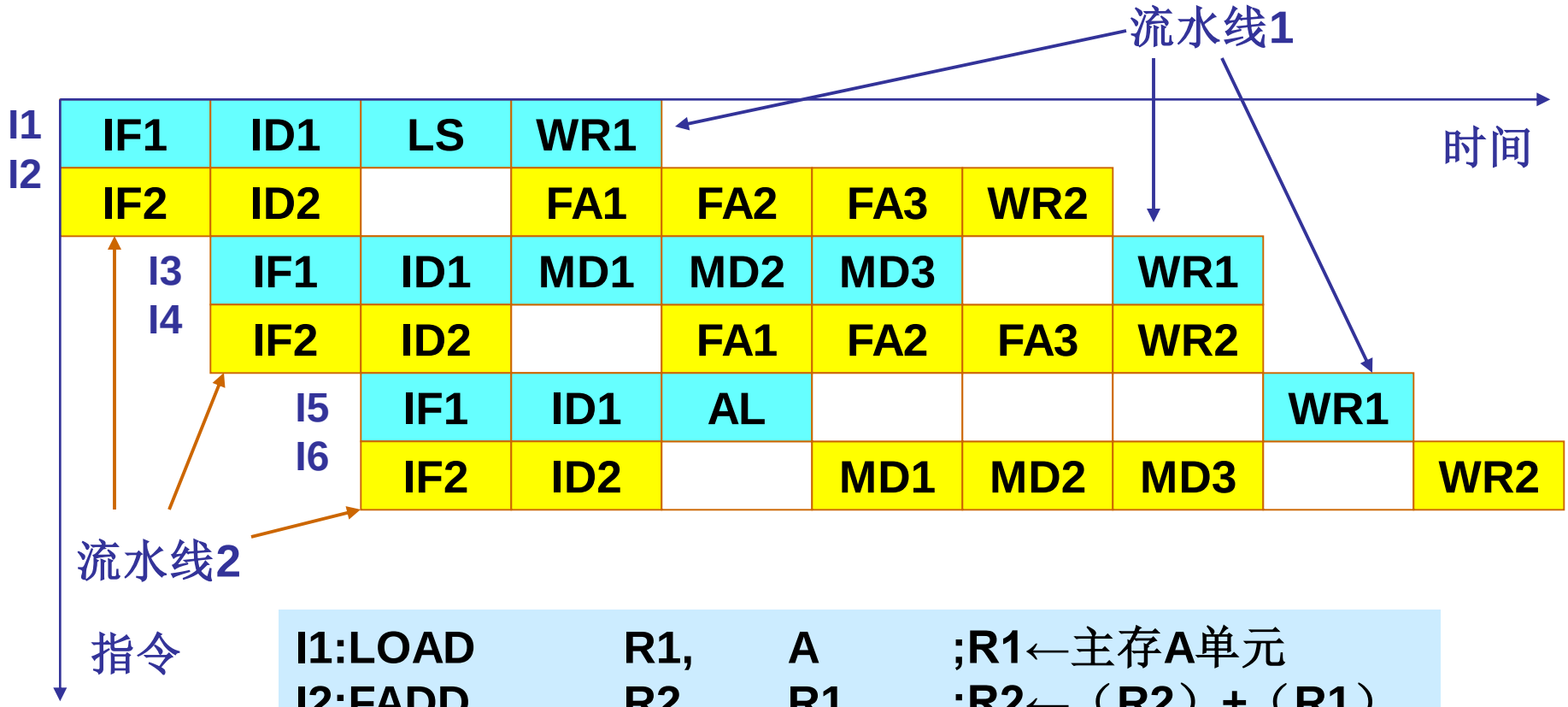
顺序（或乱序）是相对于程序顺序而言。

程序：

```
I1:LOAD   R1,   A    ;R1←主存A单元
I2:FADD   R2,   R1   ;R2←（R2）+（R1）
I3:FMUL   R3,   R4   ;R3←（R3）×（R4）
I4:FADD   R4,   R5   ;R4←（R4）+（R5）
I5:DEC    R6         ;R6←（R6）-1
I6:FMUL   R6,   R7   ;R6←（R6）×（R7）
```

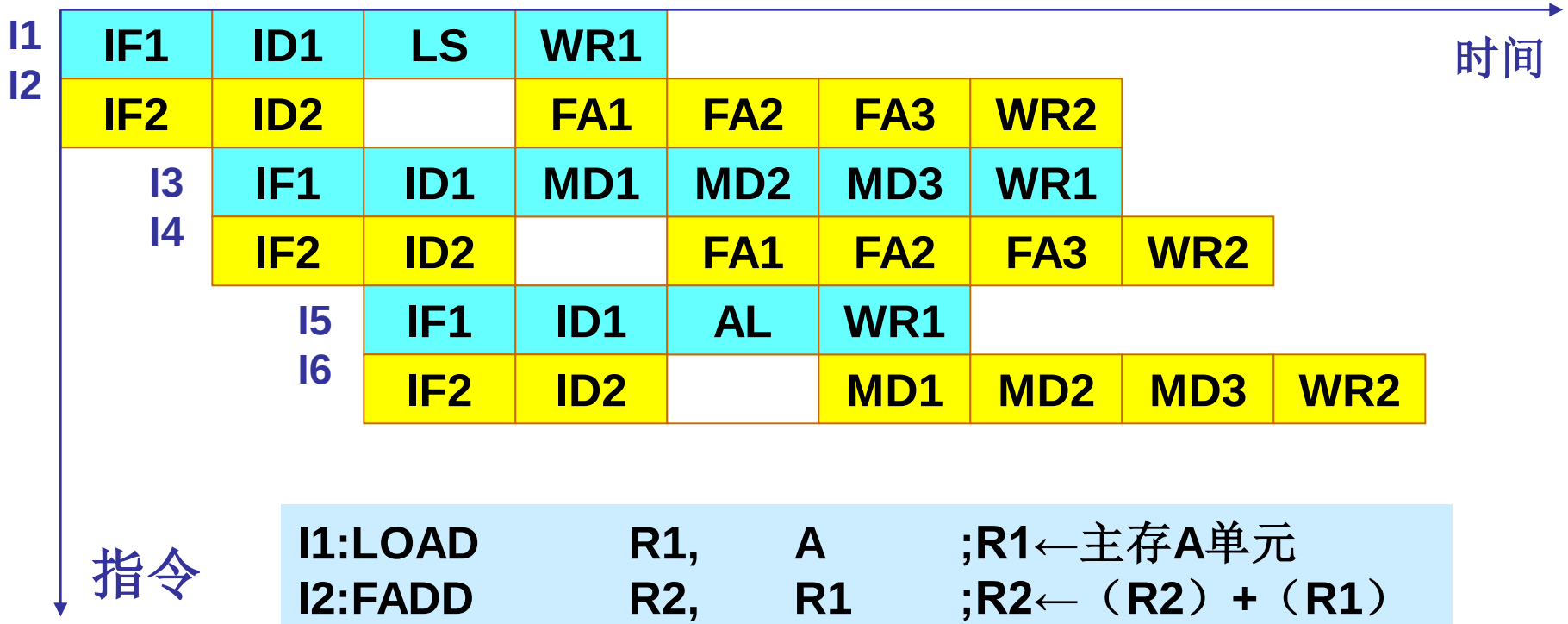


“顺序发射顺序完成”




```
I1:LOAD    R1,    A     ;R1←主存A单元
I2:FADD    R2,    R1    ;R2←（R2）+（R1）
I3:FMUL    R3,    R4    ;R3←（R3）×（R4）
I4:FADD    R4,    R5    ;R4←（R4）+（R5）
I5:DEC     R6           ;R6←（R6）-1
I6:FMUL    R6,    R7    ;R6←（R6）×（R7）
```

“顺序发射乱序完成”




```
I1:LOAD   R1,   A    ;R1←主存A单元
I2:FADD   R2,   R1   ;R2←（R2）+（R1）
I3:FMUL   R3,   R4   ;R3←（R3）×（R4）
I4:FADD   R4,   R5   ;R4←（R4）+（R5）
I5:DEC    R6         ;R6←（R6）-1
I6:FMUL   R6,   R7   ;R6←（R6）×（R7）
```

3. 超标量处理机性能

- 在指令级并行度为(m,1)、 k段流水线的超标量处理机上，执行N条指令所用的时间为：

$$T(m,1)=(k+\frac{N-m}{m})\Delta t$$

- 超标量处理机相对于单流水线普通标量处理机的加速比为：

$$S(m,1)=\frac{T(1,1)}{T(m,1)}=\frac{(k+N-1)\Delta t}{(k+\frac{N-m}{m})\Delta t}=\frac{\mathrm{m(k+N-1)}}{\mathrm{mk+N-m}}$$

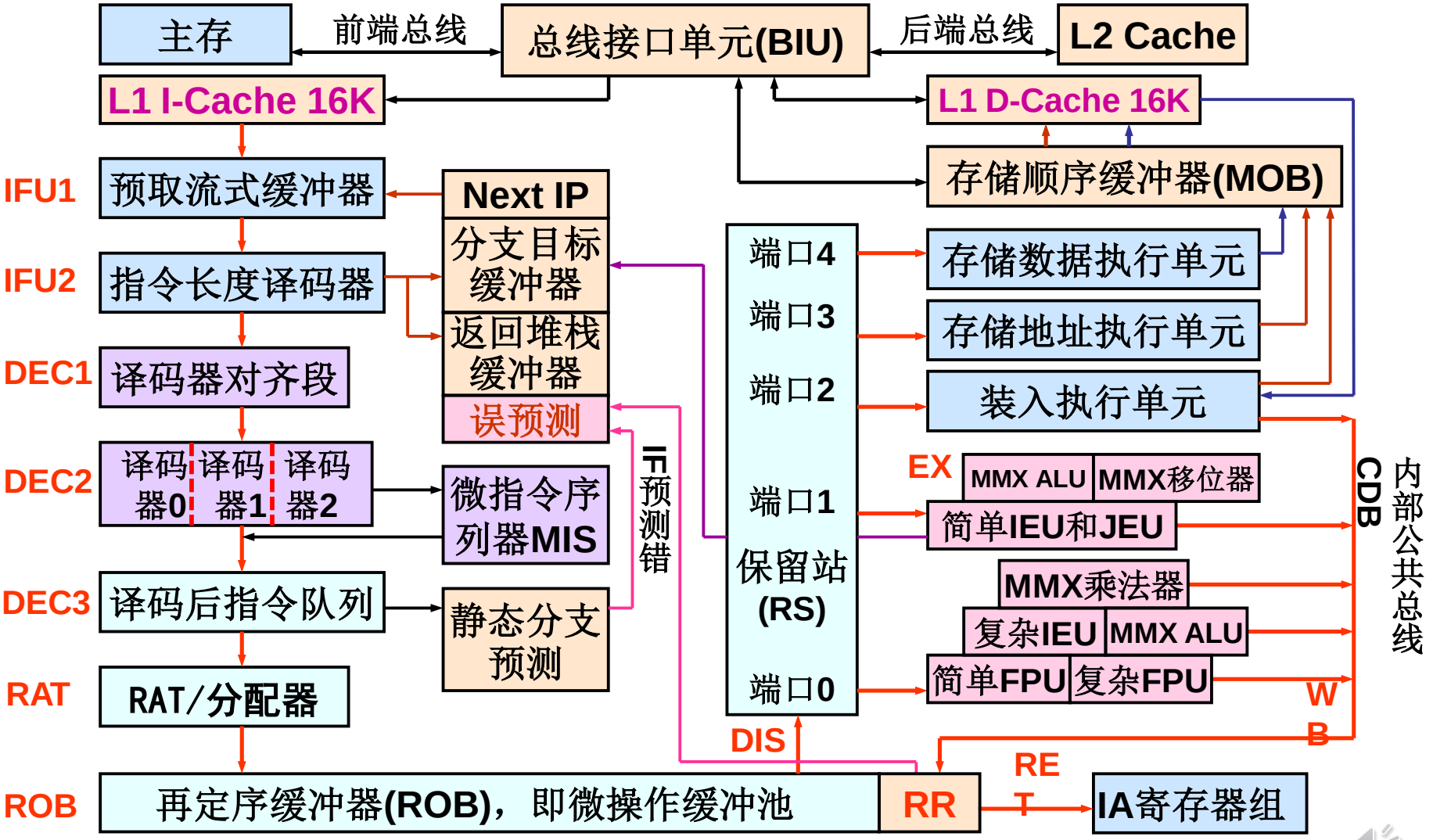
当$N\to\infty$ 时，在没有资源冲突、没有数据相关和控制相关的理想情况下，超标量处理机相对于单流水线普通标量处理机的加速比最大值为：$S(m,1)_{max}=m$

5.4.1 超标量处理机

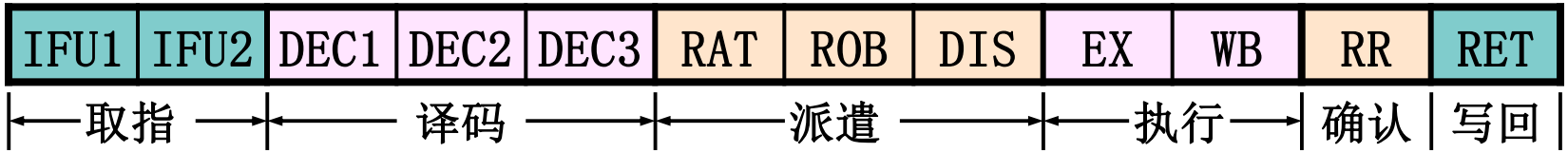
- 典型的超标量流水线处理机有IBM RS/6000、DEC 21064、Intel i960CA、Tandem、Cyclone等。
- 1986年的Intel i960CA时钟频率为25 MHz，度m=3，有7个功能部件可以并发使用。1990年的IBM RS/6000使用1μm CMOS工艺，时钟频率为30 MHz。处理机中有转移处理、定点、浮点3种功能部件，它们可并行工作。转移处理部件每Δt可执行多达5条指令，度m=4，性能可达34 MIPS和11 MFLOPS。非常适合于在数值计算密集的科学工程上应用及在多用户商用环境下工作。许多基于RS/6000的工作站和服务器都是IBM生产的。如POWER Station 530。
- 1992年的DEC 21064使用0.75μm CMOS，时钟频率为150 MHz，度m=2，10段流水线，最高性能可达300MIPS和150MFLOPS。Tandem公司的Cyclone(旋风)计算机由4到16台超级标量流水处理机组成。每个处理机的寄存器组有9个端口(其中5个为读，4个为写)，有两个算术逻辑部件，度m=2。由于程序中可开发的指令并行性有限，所以超标量流水线处理机的度m比较低。

PII CPU的超标量流水技术

特征—哈佛结构+DIB，3路超标量，动态执行技术



超标量流水线组成—12个段，以再定序缓冲器ROB为核心



⑴**IFU1 取指单元段1**

每次从L1-I$取出一个块(32B)，装入预取流式缓冲器(S=2个块)

预取流式缓冲器的空闲≥1个块时

⑵**IFU2 取指单元段2**

每次从预取流式缓冲器取出16B信息(起始位置任意)；

对16B信息预译码、标志指令边界(3条指令)；

若发现转移指令，则进行动态分支预测(指令地址送BTB)

指令被取走时

※取指段特征—按需动作(非按拍动作)，按序流动

⑶DEC1 译码段1

每次从IF段取3条IA指令，按一定次序旋转(对应译码器结构)

⑷DEC2 译码段2

每次同时译码3条指令，形成≤6个uop(118 bit/uop)

↓ ↓

(结构为复杂/简单/简单)　　(4+1+1，复杂指令通过MIS翻造)

⑸DEC3 译码段3

每次接收≤6个uop，按程序顺序排队(DIQ)；

若发现转移型uop、且BTB缺失，则进行静态分支预测；

(含IF段误预测修正)

DIQ空闲≥6个uop时

※译码段特征—CISC→RISC，按需动作，按序流动

⑹RAT 寄存器别名表和分配器段 --按序发射

每拍取3个uop，将uop中的IA寄存器，转换为内部寄存器

(如ROB“目的值”字段)←┘

△此段消除/转化了WAR冒险、消除了WAW冒险

⑺ROB 再定序缓冲器段

每拍接收3个uop、按序存放在ROB中

△ROB为环形缓冲区，管理各个uop(收数据、改状态)

⑻DIS 派遣段 --乱序派遣

每拍RS从ROB以次序任意拷贝多个OPD就绪的uop到相应端口；

RS在EX单元可用时，发送相应端口的uop至EX单元

△此段消除了RAW冒险

※派遣段特征—冒险处理，按需动作→按拍动作，

按序流动→乱序流动

⑼EX 执行段

各部件执行uop，结果送上CDB；若uop为分支操作，更新BTB

⑽WB 写回段

ROB从CDB接收执行结果、修改状态

※执行段特征—乱序执行，部件时延可不等(动态流水线)

⑾RR 回收就绪段(确认段)

按程序顺序、以IA指令为单位，对所含uop进行确认；

处理分支误预测、异常(清除部分或全部ROB)

※确认段特征—RISC→CISC，乱序流动→按序流动

⑿RET 回收段

将IA指令的结果写回IA寄存器，或通知MOB完成写L1$；

清除ROB中该IA指令对应的uop(有效位复位)

5.4.2 超流水线处理机

- **两种定义：**
  - 一个周期内能够**分时发射**多条指令的处理机称为超流水线处理机。
  - 指令流水线有8个或更多功能段的流水线处理机称为超流水线处理机。

5.4.2 超流水线处理机

- 不同于超标量处理机和超长指令字处理机，超流水线处理机每个△t’仍只流出一条指令，但它的△t’很小
- 一台度为n的超流水线处理机，△t’只是基本机器周期△t的1/n
- 因此，只要流水线的性能得以发挥，其并行度就可以达到n

提高处理机性能的不同方法

- **超标量处理机利用资源重复**，设置多个执行部件寄存器端口
- **超流水线处理机则侧重开发时间并行性**，在公共硬件上采用较短的时钟周期、深度流水来提高速度

提高处理机性能的不同方法

超标量处理机是通过增加硬件资源为代价来换取处理机性能的。
超流水线处理机则通过各硬件部件充分重叠工作来提高处理机性能。

两种不同并行性：
超标量处理机采用的是空间并行性。
超流水线处理机采用的是时间并行性。

提高处理机性能的不同方法

空间并行性：

设置多个独立的操作部件

多操作部件处理机

超标量处理机

时间并行性：

采用流水线技术。

不增加或只增加少量硬件就能使运算速度提高几倍

流水线处理机

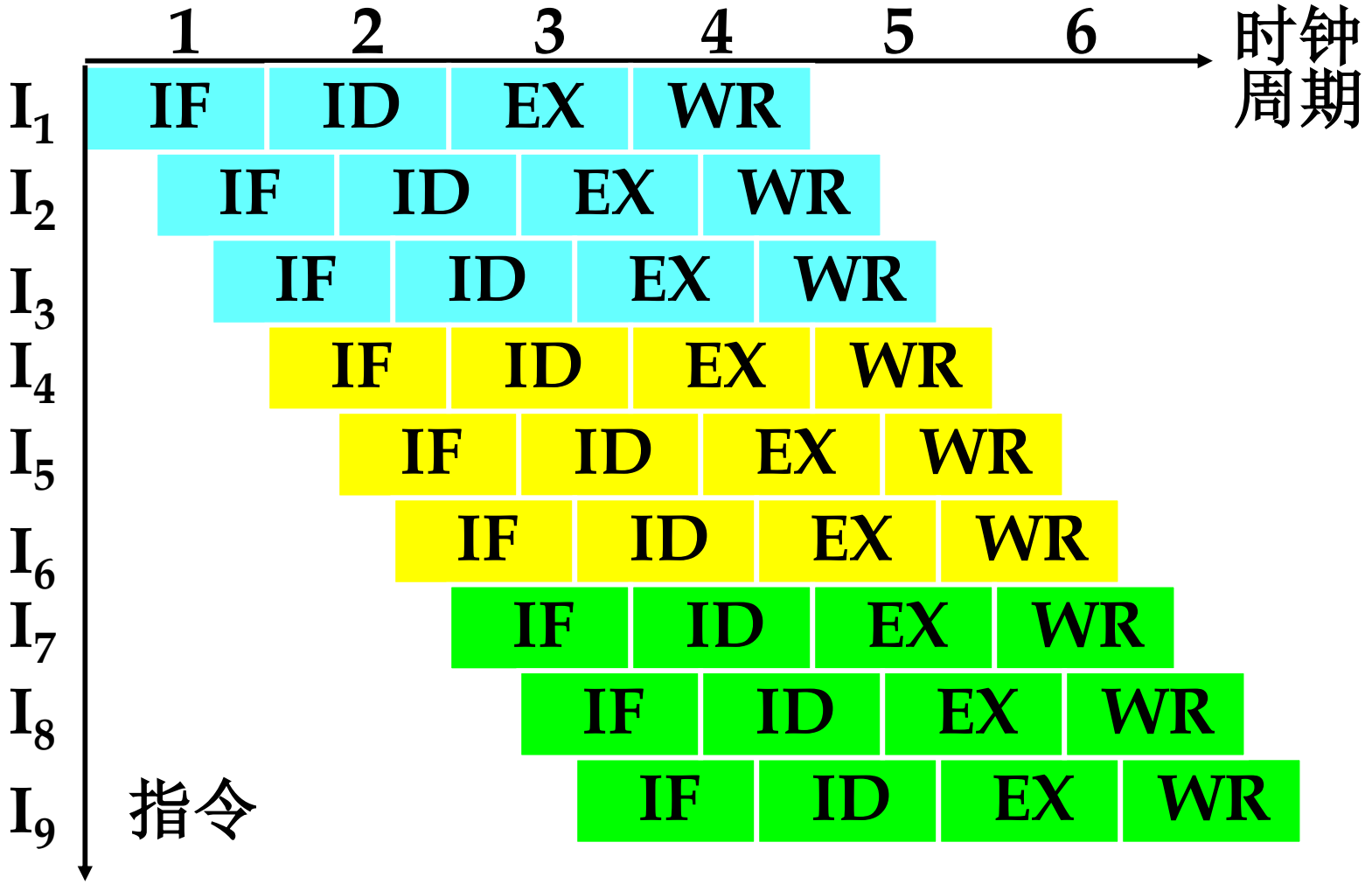
超流水线处理机

超流水线处理机指令执行序列

- 超流水线处理机需要使用多相时钟
- 没有高速时钟机制，超流水线处理机是无法实现的
- 指令执行时序
  - 每隔1/n个时钟周期发射一条指令，流水线周期为1/n个时钟周期

超流水线处理机指令执行序列

每个时钟周期分时发送3条指令的超流水线



超流水线处理机性能

- 在指令级并行度为(1, n)、k段流水线的超流水线处理机执行N条指令所的时间为：

$$T(1,n)=(k+\frac{N-1}{n})\Delta t$$

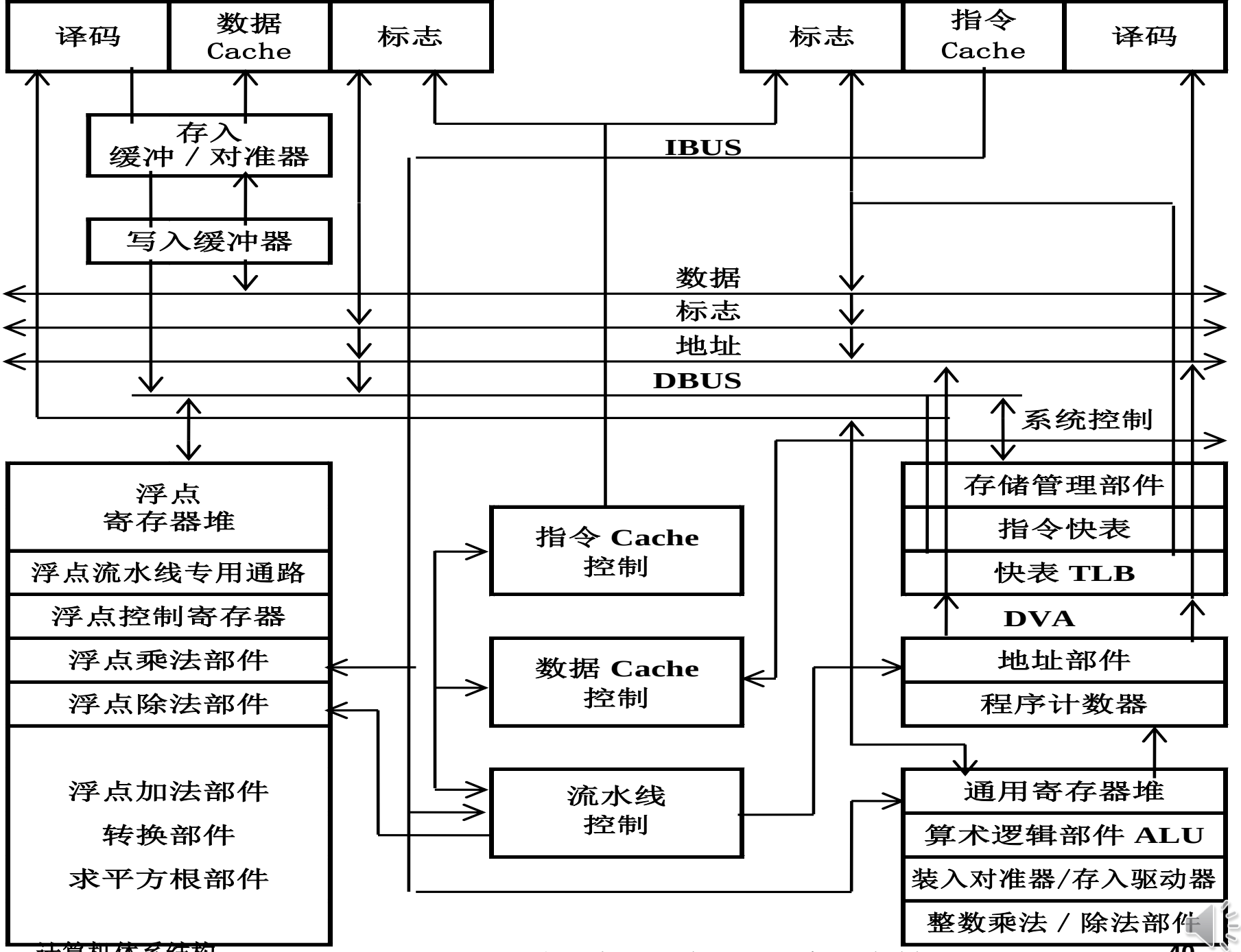
- 超流水线处理机相对于单流水线普通标量处理机的加速比为：

$$S(1,n)=\frac{T(1,1)}{T(1,n)}=\frac{(k+N-1)\Delta t}{(k+\frac{N-1}{n})\Delta t}=\frac{n(k+N-1)}{nk+N-1}$$

当$N\to\infty$ 时，在没有资源冲突、没有数据相关和控制相关的理想情况下，超流水线处理机相对于单流水线普通标量处理机的加速比最大值为：$S(1,n)_{max}=n$

超流水线处理机： MIPS R4000

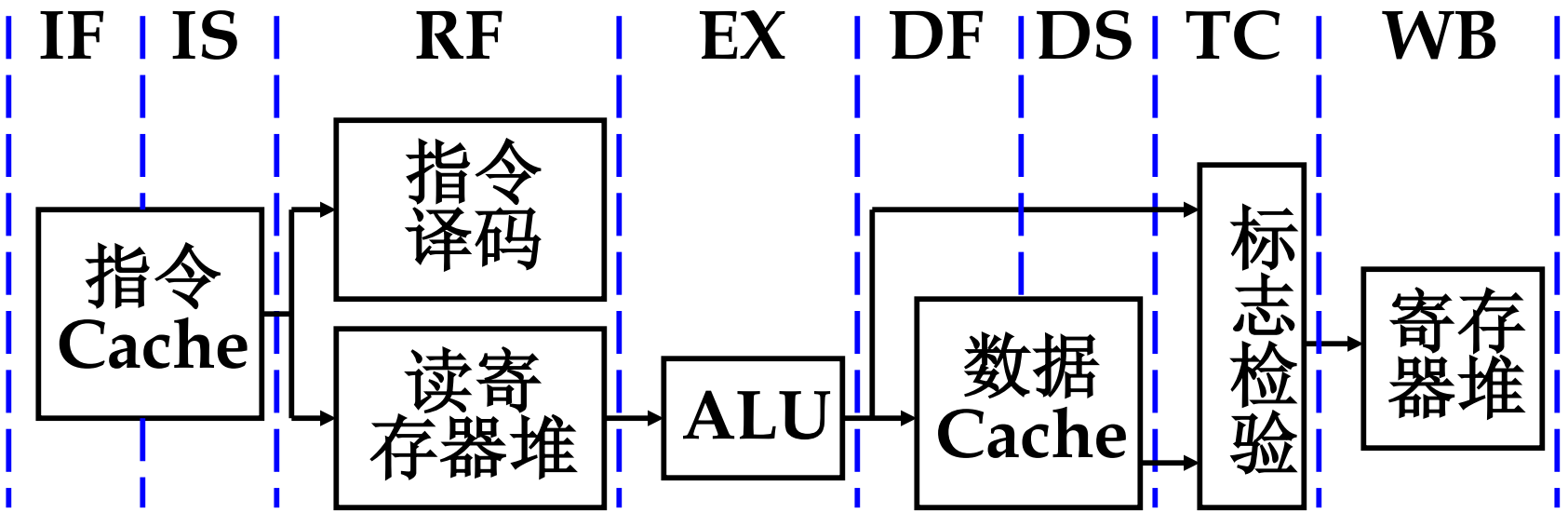
- 每个时钟周期包含两个流水段，是一种很标准的超流水线处理机结构。指令流水线有8个流水段。
- 有两个Cache：指令Cache和数据Cache的容量各8KB。每个时钟周期可以访问Cache两次，因此在一个时钟周期内可以从指令Cache中读出两条指令，从数据Cache中读出或写入两个数据。
- 主要运算部件有整数部件和浮点部件。



MIPS R4000 超流水线处理机结构

超流水线处理机： MIPS R4000

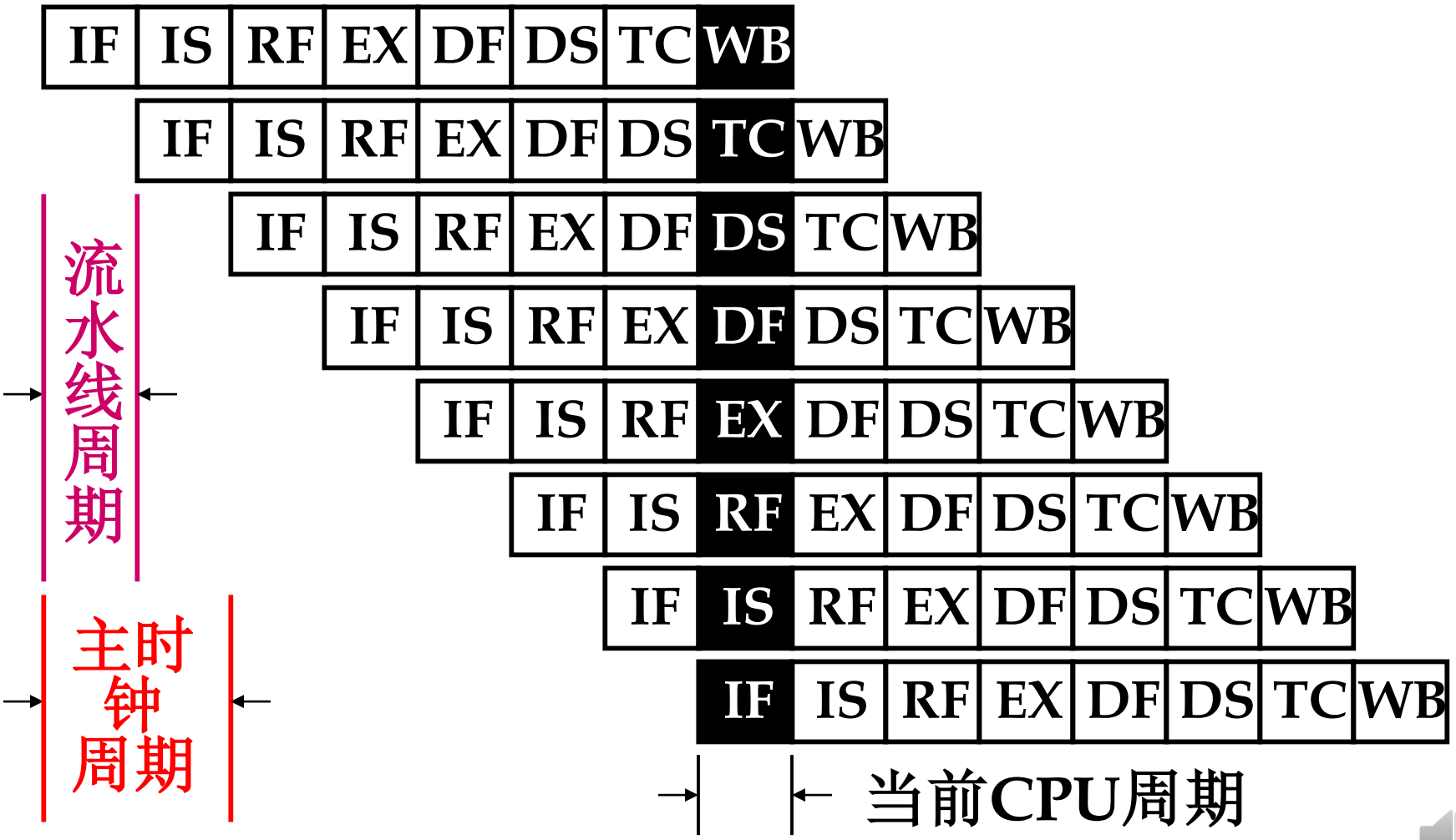
MIPS R4000处理机的流水线操作



IF：取第一条指令　　IS：取第二条指令
RF：读寄存器堆，指令译码
EX：执行指令　　DF：取第一个数据
DS：取第二个数据　　TC：数据标志校验；
WB：写回结果

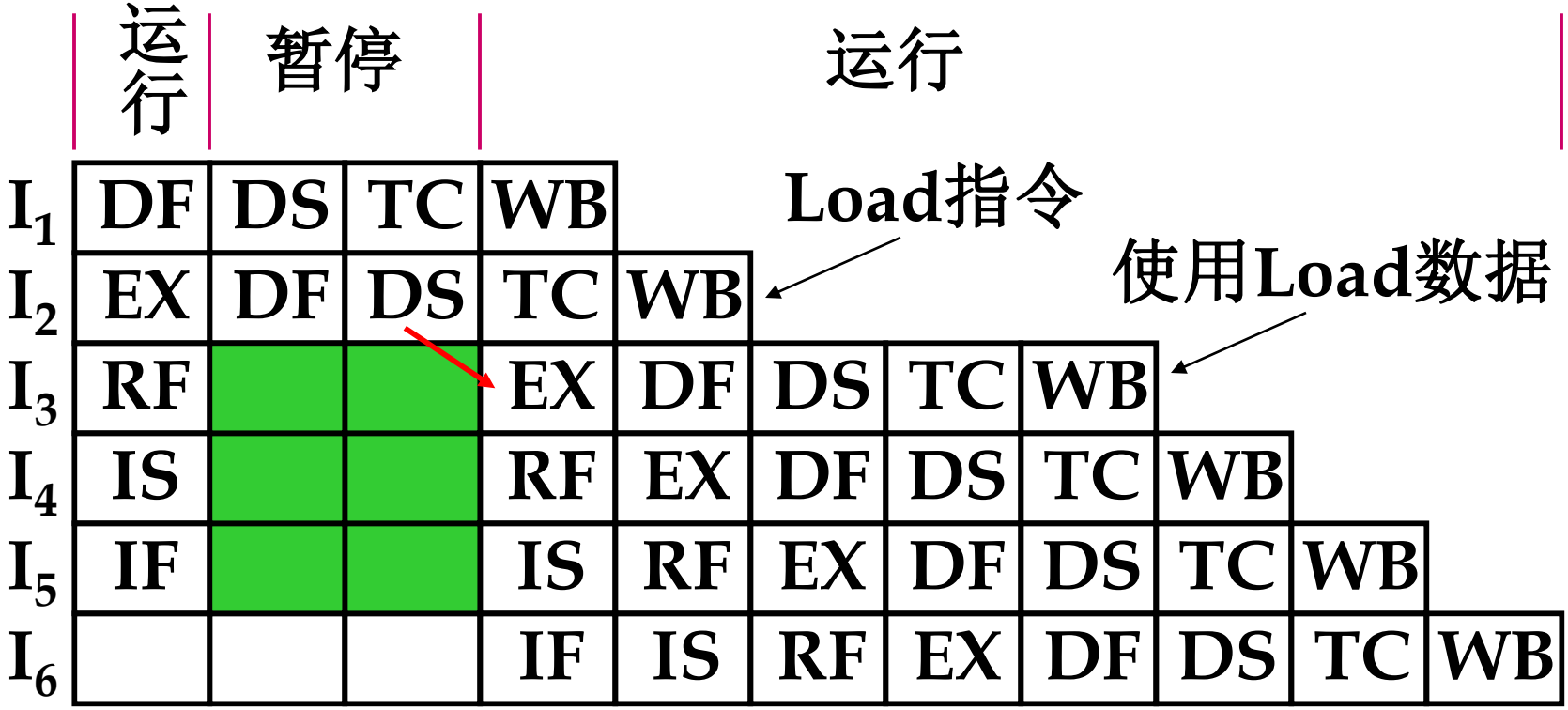
超流水线处理机： MIPS R4000

MIPS R4000正常指令流水线工作时序



超流水线处理机： MIPS R4000

如果在LOAD指令之后的两条指令中，任何一条指令要在它的EX流水级使用这个数据，则指令流水线要暂停一个时钟周期。 采用顺序发射方式。

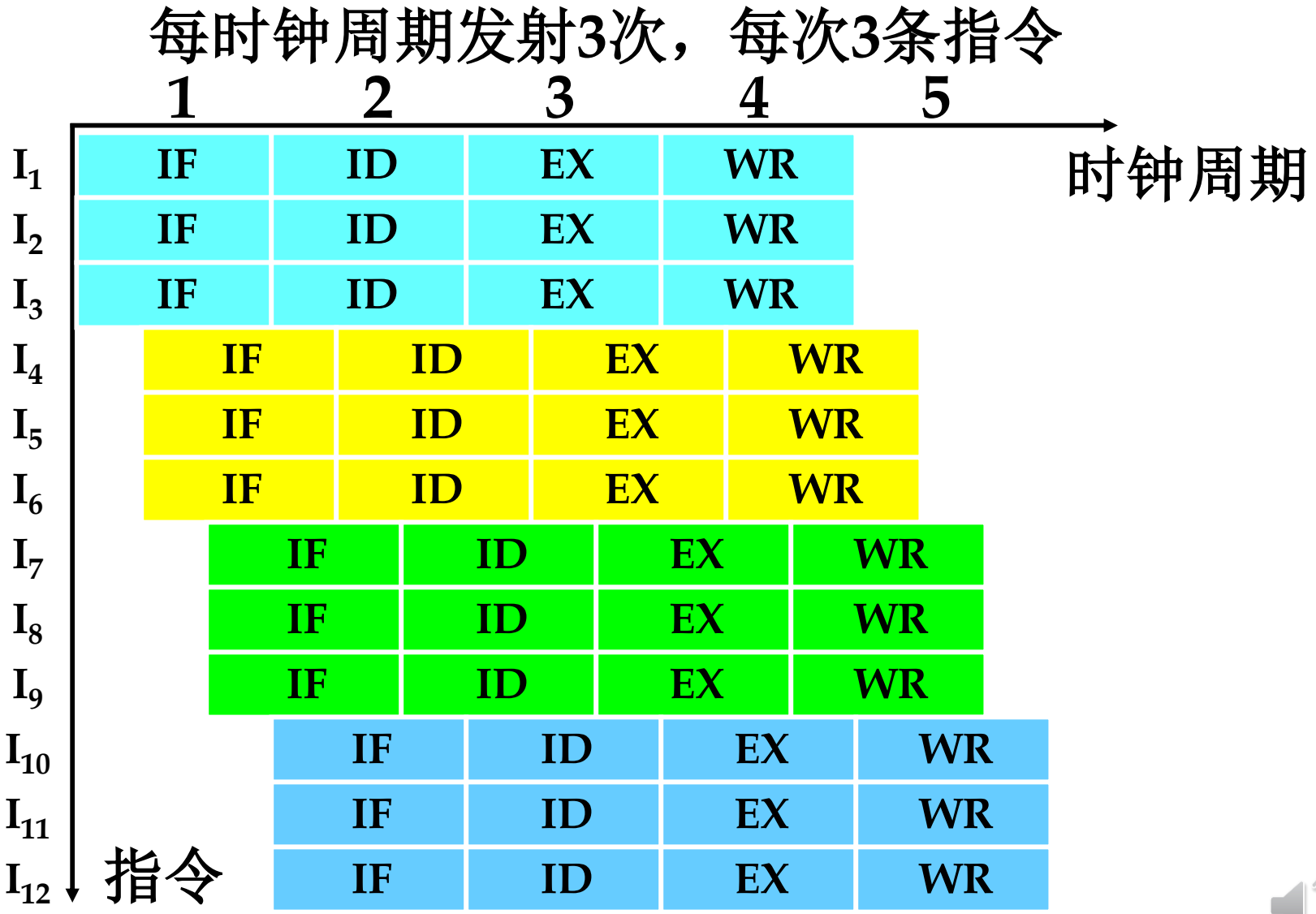


MIPS R4000 有暂停的指令流水线工作时序

5.4.3 超标量超流水线处理机

- 把超标量与超流水线技术结合在一起，就成为**超标量超流水线处理机。**
- 指令执行时序
  - 超标量超流水线处理机在一个时钟周期内分时发射指令n次，每次同时发射指令m条，每个时钟周期总共发射指令$m \cdot n$条。

指令执行序列


每时钟周期发射3次，每次3条指令
1
2
3
4
5
时钟周期
I1 IF ID EX WR
I2 IF ID EX WR
I3 IF ID EX WR
I4 IF ID EX WR
I5 IF ID EX WR
I6 IF ID EX WR
I7 IF ID EX WR
I8 IF ID EX WR
I9 IF ID EX WR
I10 IF ID EX WR
I11 IF ID EX WR
I12 IF ID EX WR
指令

超标量超流水线处理机性能

- 指令级并行度为(m，n)的超标量超流水线处理机，连续执行N条指令所需要的时间为：

$$T(m,n)=(k+\frac{N-m}{m\cdot n})\Delta t$$

- 超标量超流水线处理机相对于单流水线标量处理机的加速比为：

$$S(m,n)=\frac{T(1,1)}{T(m,n)}=\frac{(k+N-1)\Delta t}{(k+\frac{N-m}{mn})\Delta t}=\frac{\mathrm{mn(k+N-1)}}{\mathrm{mnk+N-m}}$$

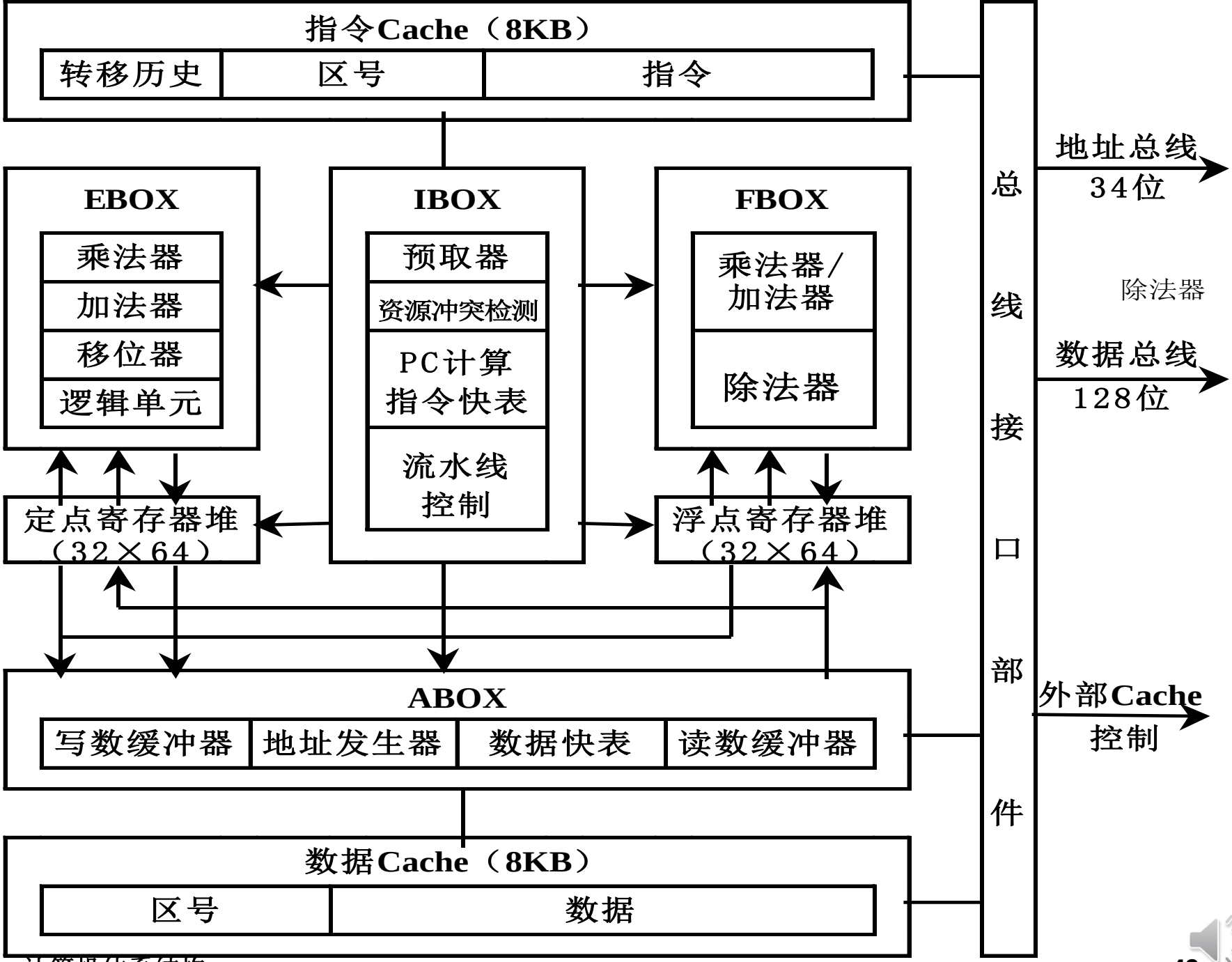
当$N\to\infty$ 时，在没有资源冲突、没有数据相关和控制相关的理想情况下，超标量超流水线处理机相对于单流水线普通标量处理机的加速比最大值为：$S(m,n)_{max}=m\times n$

超标量超流水线处理机： DEC Alpha

- 主要由四个功能部件和两个Cache组成：整数部件EBOX、浮点部件FBOX、地址部件ABOX和中央控制部件IBOX。
- 中央控制部件IBOX可以同时从指令Cache中读入两条指令，同时对读入的两条指令进行译码，并且对这两条指令作资源冲突检测，进行数据相关性和控制相关性分析。如果资源和相关性允许，IBOX就把两条指令同时发射给EBOX、ABOX和FBOX三个指令执行部件中的两个。
- 指令流水线采用顺序发射乱序完成的控制方式。在指令Cache中有一个转移历史表，实现条件转移的动态预测。在EBOX内还有多条专用数据通路，可以把运算结果直接送到执行部件。

超标量超流水线处理机： DEC Alpha

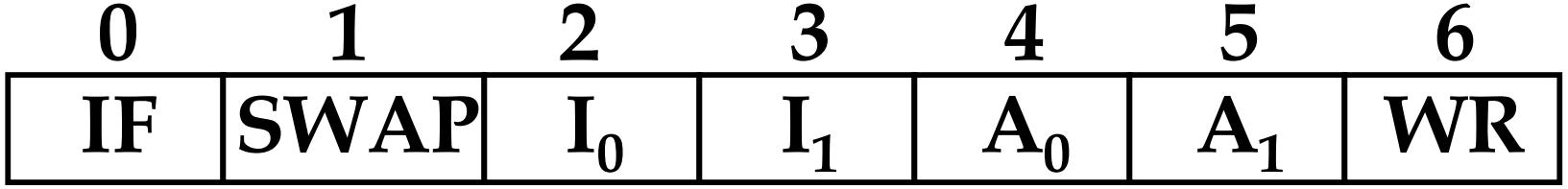
- **Alpha 21064处理机共有三条指令流水线**
  整数操作流水线和访问存储器流水线分为7个流水段，其中，取指令和分析指令为4个流水段，运算2个流水段，写结果1个流水段。浮点操作流水线分为10个流水段，其中，浮点执行部件FBOX的延迟时间为6个流水段。
- **所有指令执行部件EBOX、IBOX、ABOX和FBOX中都设置由专用数据通路。**
- **Alpha 21064处理机的三条指令流水线的平均段数为8段，每个时钟周期发射两条指令。因此，Alpha 21064处理机是超标量超流水线处理机。**



Alpha 21064 处理机结构

超标量超流水线处理机： DEC Alpha

7个流水段的整数操作流水线



IF 取值
SWAP 交换双发射指令、转移预测
I_0 指令译码
I_1 访问通用寄存器堆，发射校验
A_1 计算周期1，IBOX计算新的PC值
A_2 计算周期2，查指令快表
WR 写整数寄存器堆，指令Cache命中检测

超标量超流水线处理机： DEC Alpha

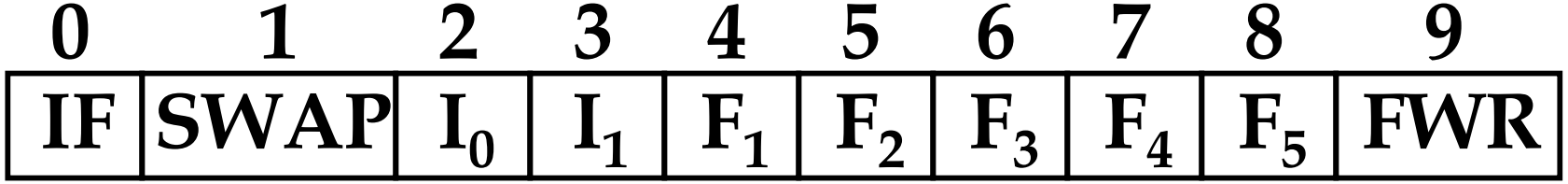
7个流水段的访问存储器流水线

0	1	2	3	4	5	6
IF	SWAP	I_0	I_1	AC	TB	HM

IF 取值
SWAP 交换双发射指令、转移预测
I_0 指令译码
I_1 访问通用寄存器堆，发射校验
AC ABOX计算有效数据地址
TB 查数据快表
HM 写读数缓冲栈，数据Cache命中/不命中检测

超标量超流水线处理机： DEC Alpha

10个流水段的浮点操作流水线



IF 取值

SWAP 交换双发射指令、转移预测

I_0 指令译码

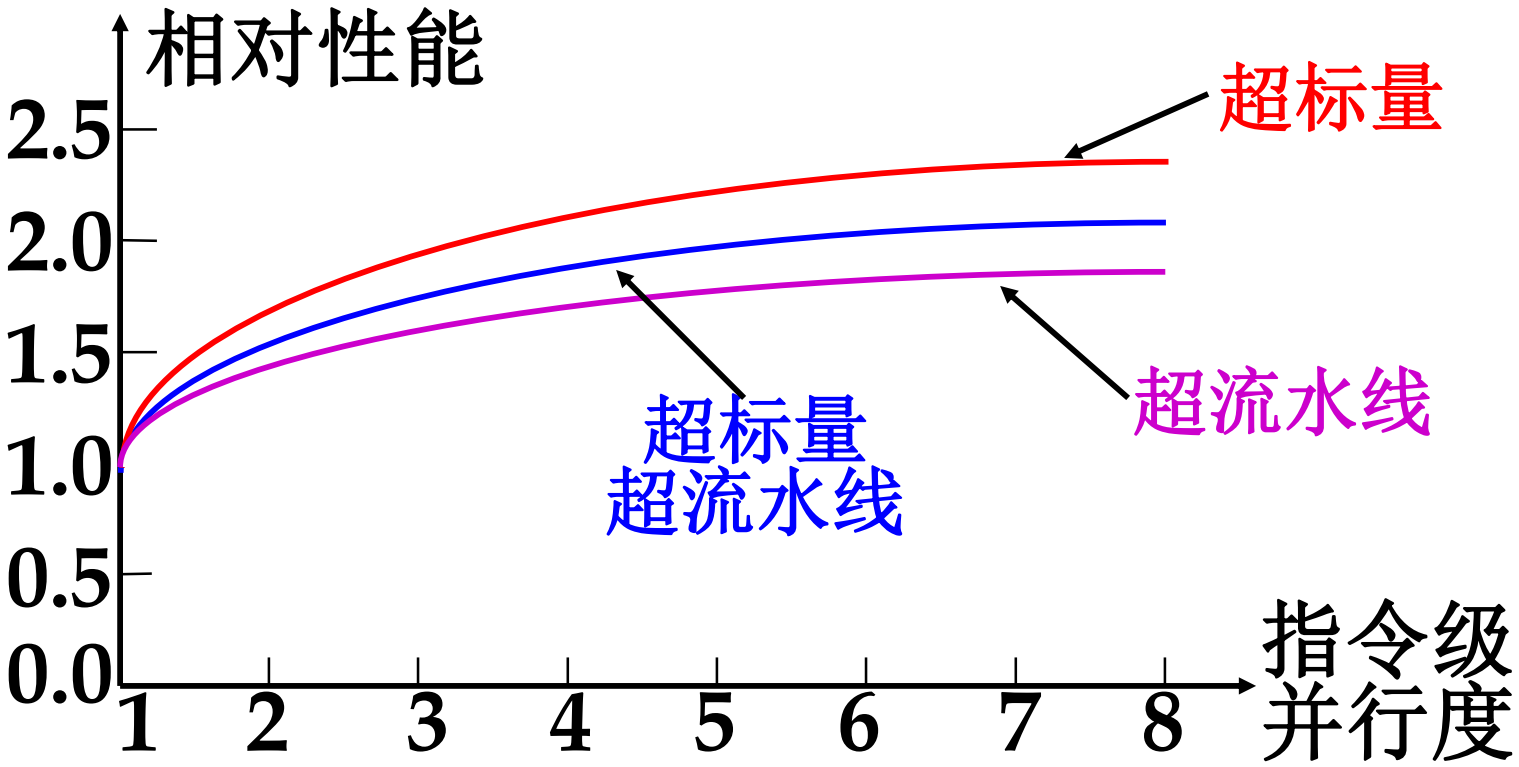
I_1 访问通用寄存器堆，发射校验

F_1-F_5 浮点计算流水线

FWR 写回浮点寄存器堆

5.4.4 三种指令级并行处理机性能比较

性能比较



5.4.4 三种指令级并行处理机性能比较

- **性能比较**
  - 最高：超标量处理机；
  - 其次：超标量超流水线处理机；
  - 最低：超流水线处理机。

主要原因：

(1)超流水线处理机的启动延迟比超标量处理机大。

(2)条件转移造成的损失，超流水线处理机要比超标量处理机大。

(3)超标量处理机指令执行部件的冲突要比超流水线处理机小。

5.4.4 三种指令级并行处理机性能比较

- 实际指令级并行度与理论指令级并行度的关系
  - 当横坐标给出的理论指令级并行度比较低时，处理机的实际指令级并行度的提高比较快。
  - 当理论指令级并行度进一步增加时，处理机实际指令级并行度提高的速度越来越慢。
  - 在实际设计超标量、超流水线、超标量超流水线处理机的指令级并行度时要适当，否则，有可能造成花费了大量的硬件，但实际上处理机所能达到的指令级并行度并不高。
  - 目前，一般认为，m 和 n 都不要超过4。

5.4.4 三种指令级并行处理机性能比较

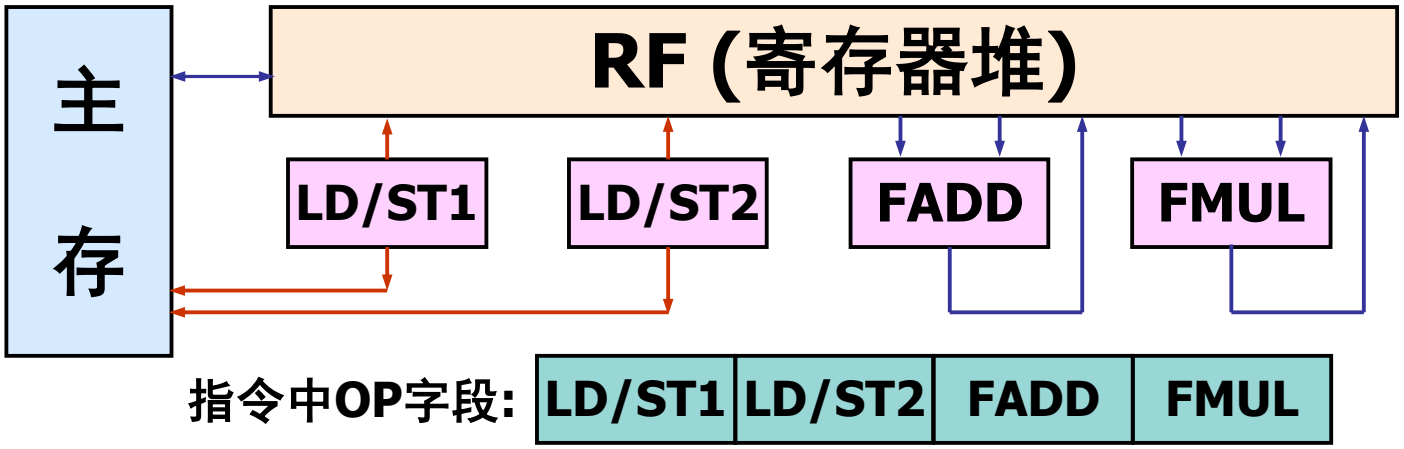
- 最大指令级并行度
  - 一个特定程序由于受到本身的数据相关和控制相关的限制，它的指令级并行度的最大值是有限的，是有个确定的值。
  - 这个最大值主要由程序自身的语义来决定，与这个程序运行在那一种处理机上无关。
  - 对于某一个特定的程序，图中的三条曲线最终都要收拢到同一个点上。当然，对于各个不同程序，这个收拢点的位置也是不同的。

5.4.5 超长指令字处理机

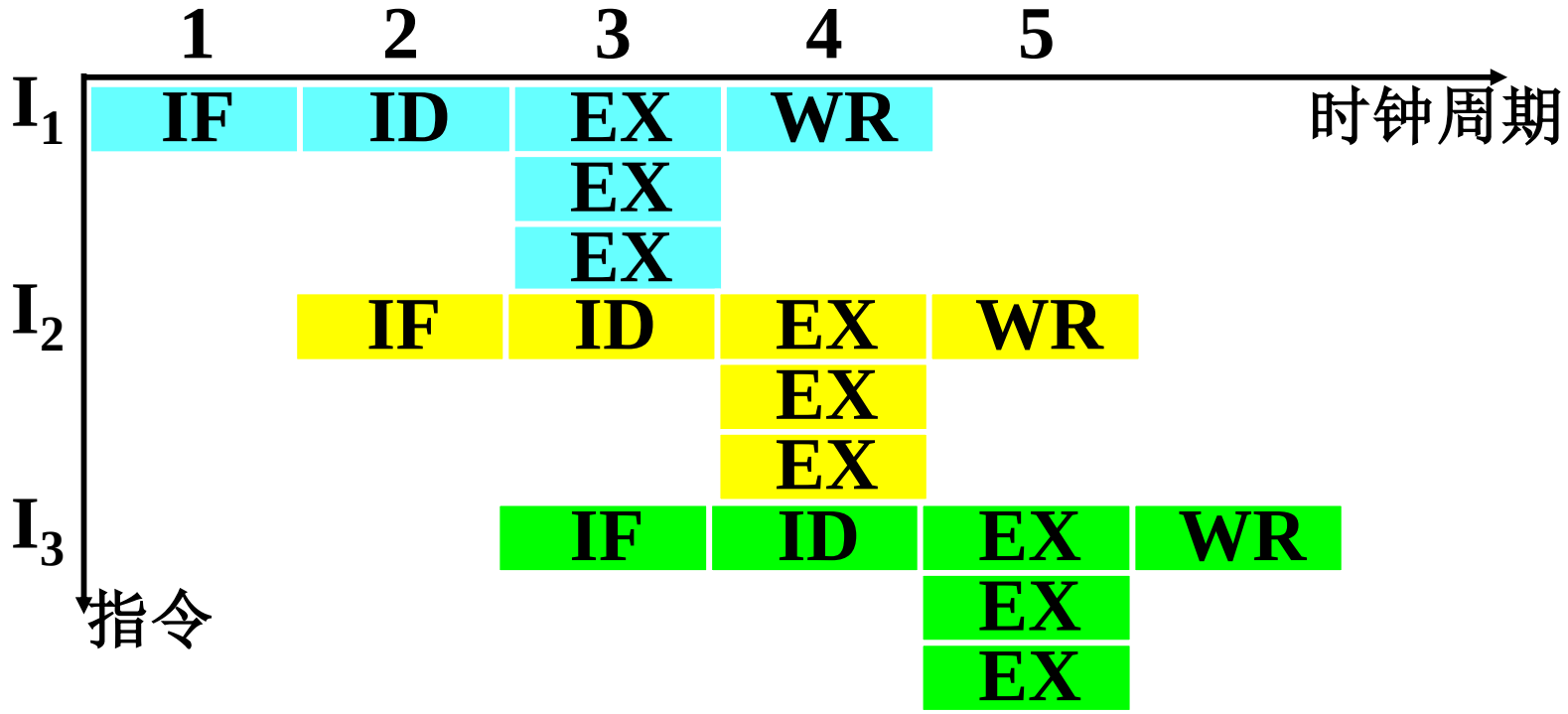
- 1983年，Yale大学Fisher教授首先提出。
- VLIW (Very Long Instruction Word) 结构将水平型微码与超标量处理两者相结合，指令字长可达数百位，多个功能部件并发工作，共享大容量寄存器堆
- 与超标量处理机不同的是，编译时，编译程序找出指令间潜在的并行性，将多个能并行执行的不相关或无关的操作先行压缩，组合在一起，形成一条有多个操作段的超长指令
- 运行时，这条超长指令控制多个相互独立的功能部件并行操作

5.4.5 超长指令字处理机

- 特点：
  - 指令字长很长，可达数百位；
  - 有多个功能部件并发工作；
  - 用一条长指令来实现多个操作的并行执行；



5.4.5 超长指令字处理机



超长指令字处理机流水线时空图

每拍启动1条指令，要求并行度=3

5.4.2 超长指令字处理机

- 优点
  - 每条指令所需拍数比超标量处理机少
  - 指令译码容易
  - 开发标量操作间的随机并行性更方便
- 缺点
  - 取决于压缩的效率
  - 其结构的目标码与一般的计算机不兼容
  - 指令字很长，但操作段各是固定，容易浪费存储空间

例题

- 在下列不同结构的处理器上运行8×8的矩阵乘法C=A×B，计算所需要的最短时间。
- 只计算乘法指令和加法指令的执行时间，不计算取操作数、数据传送和程序控制等指令的执行时间。
- 加法部件和乘法部件的延迟时间都是3个时钟周期，另外，加法指令和乘法指令还要经过一个“取指令”和“指令译码”的时钟周期，每个时钟周期为20ns，C的初始值为“0”。
- 各操作部件的输出端有直接数据通路连接到有关操作部件的输入端，在操作部件的输出端设置有足够容量的缓冲寄存器。

例题

1. 处理器内只有一个通用操作部件，采用顺序方式执行指令。
2. 单流水线标量处理器，有一条两个功能的静态流水线，流水线每个功能段的延迟时间均为一个时钟周期，加法操作和乘法操作各经过3个功能段。
3. 多操作部件处理器，处理机内有独立的乘法部件和加法部件，两个操作部件可以并行工作。只有一个指令流水线，操作部件不采用流水线结构。
4. 单流水线标量处理器，处理机内有两条独立的操作流水线，流水线每个功能段的延迟时间均为一个时钟周期。

例题

5. 超标量处理器，每个时钟周期同时发射一条乘法指令和一条加法指令，处理机内有两条独立的操作流水线，流水线的每个功能段的延迟时间均为一个时钟周期。
6. 超流水线处理器，把一个时钟周期分为两个流水级，加法部件和乘法部件的延迟时间都为6个流水级，每个时钟周期能够分时发射两条指令，即每个流水级能够发射一条指令。
7. 超标量超流水线处理器，把一个时钟周期分为两个流水级，加法部件和乘法部件延迟时间都为6个流水级，每个流水级能够同时发射一条乘法指令和一条加法指令。

例题解答

要完成两个8×8矩阵相乘，共要进行8×8×8=512次乘法， 8×8×7=448次加法。典型的C代码如下：

```
int k;
for(int i=0; i<8; i++)
    for(int j=0; j<8; j++)
    {  sum=0;
       for(k=0; k<8; k++)
       {
            sum+=A[i][k]×B[k][j]
       }
       C[i][j]=sum;
    }
```

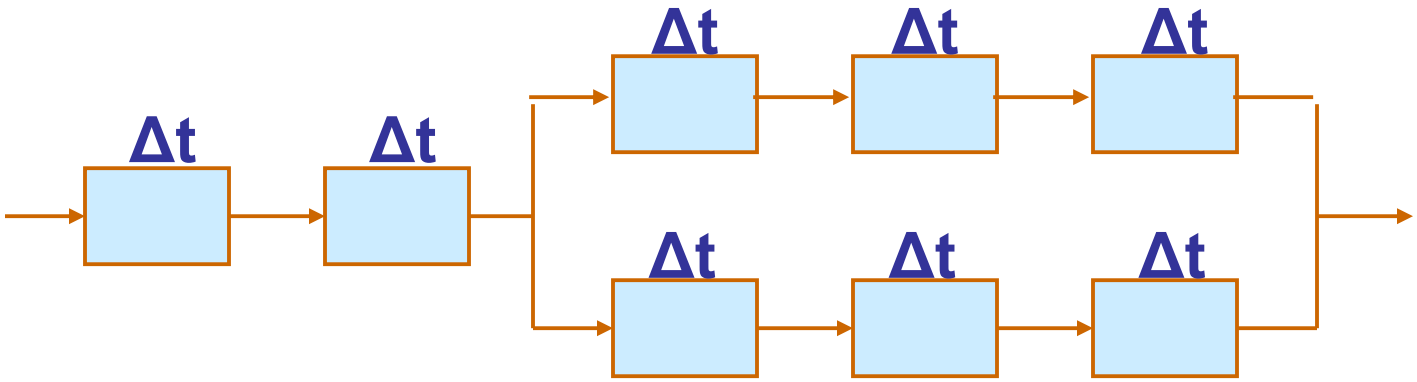
例题解答

1. 顺序执行时，每个乘法和加法指令都需要5个时钟周期，计算所需要的时间为：

   T=（512+448）×5×20ns=96000ns

2. 单流水线标量处理机，采用两功能静态流水线，结构如下。因为有足够的缓冲寄存器，所以我们可以首先把所有的乘法计算完，排空后再通过调度使加法流水线不出现停顿。计算所需要的最短时间为：

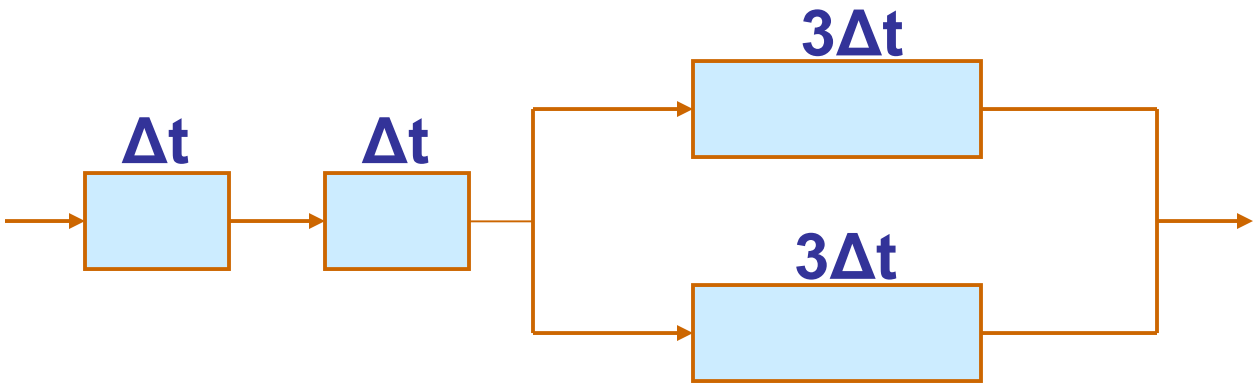
   T=[(2+3+512-1)+(3+448-1)]×20ns=19320ns



例题解答

3. 多处理部件处理机，只有一条指令流水线，结构如下图。因为加法总共执行448次，而乘法共执行512次，因此加法的执行过程一定能与乘法在某些段内并行执行，同时要考虑可能出现的乘法流水线的乘积与加法流水线的加数之间的“先写后读”数据相关。由于存在数据相关，最后一次加法运算结束与最后一次乘法运算结束的时间差为一次完整的加法流水线操作过程，为3个时钟周期。计算所需要的最短时间为：

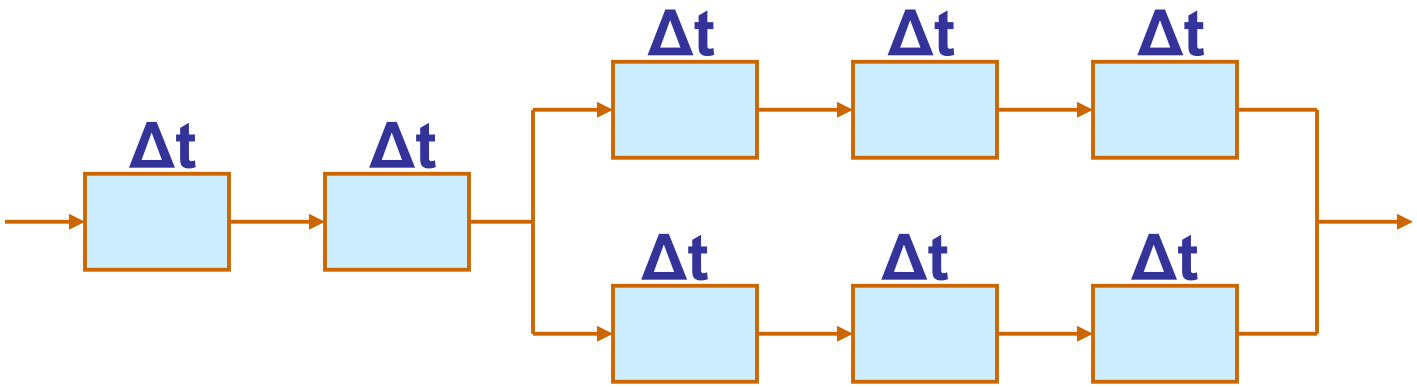
$$T=[2+3+(512-1)\times 3+3]\times 20ns=30820ns$$



例题解答

4. 单流水线标量处理机，有两条独立的操作流水线，结构如下图。（分析方法同2），但不需进行排空处理。计算所需要的最短时间为：

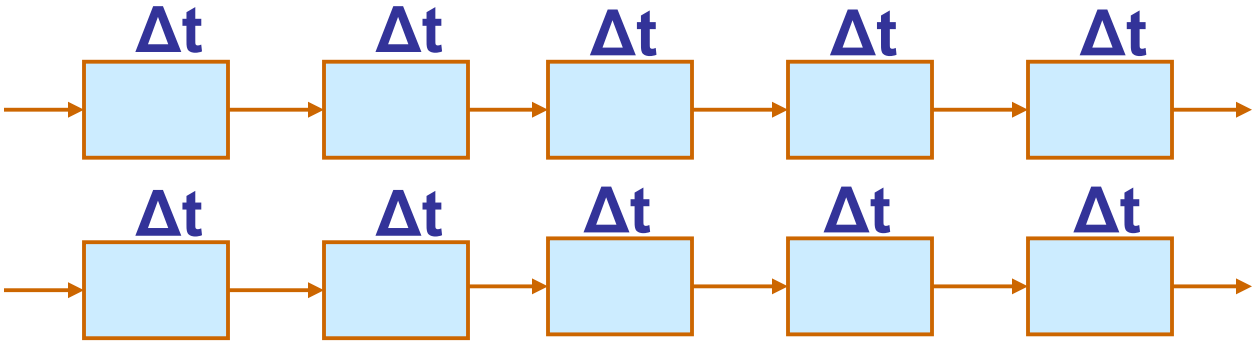
$$T=[5+(512-1)+3]\times 20ns=10380ns$$



例题解答

5. 超标量处理机，能同时发射一条加法和一条乘法指令，有两条独立的操作流水线，结构如下图。（分析方法同3），不同之处在于乘与加操作均是流水化，且是在不同的流水线上并行执行的。计算所需要的最短时间为：

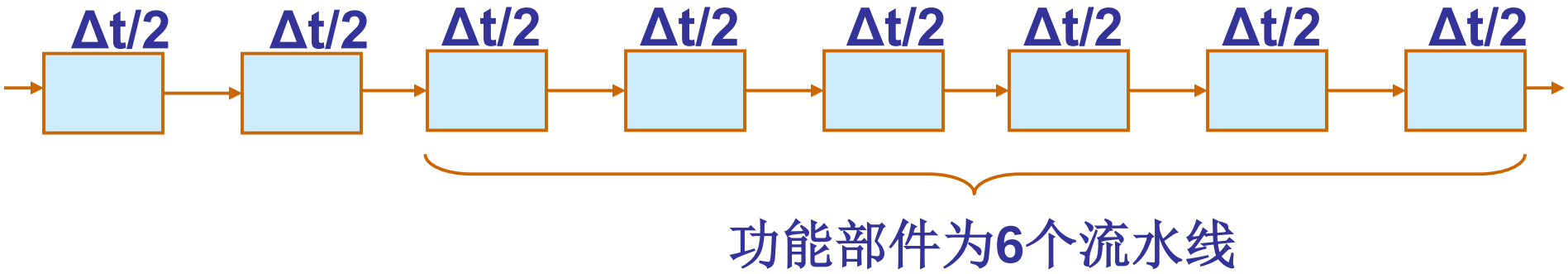
$$T=[2+3+(512-1)+448]\times 20ns=19280ns$$



例题解答

6. 超流水线处理机，每个时钟周期分时发射两条指令，加法部件和乘法部件都为6个流水线，“取指令”和“指令译码”仍分别为一个流水级，结构如下图。（分析方法同4），不同之处在于时钟周期变成了10ns，且流水线已细化。计算所需要的最短时间为：

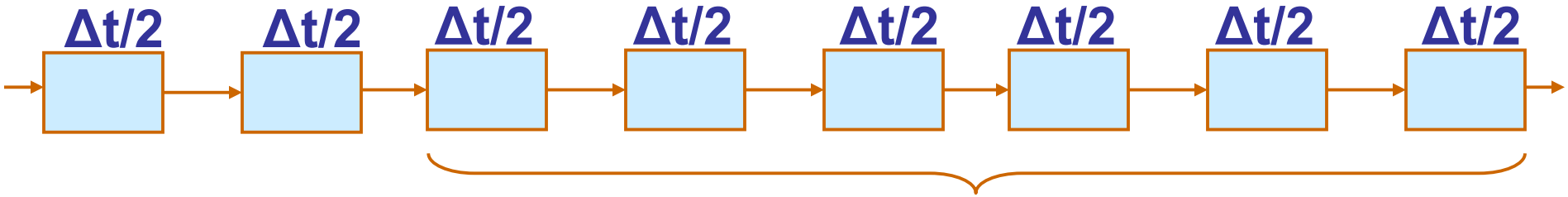
$$T=[2+6+(512-1)+448]\times 10ns=9670ns$$



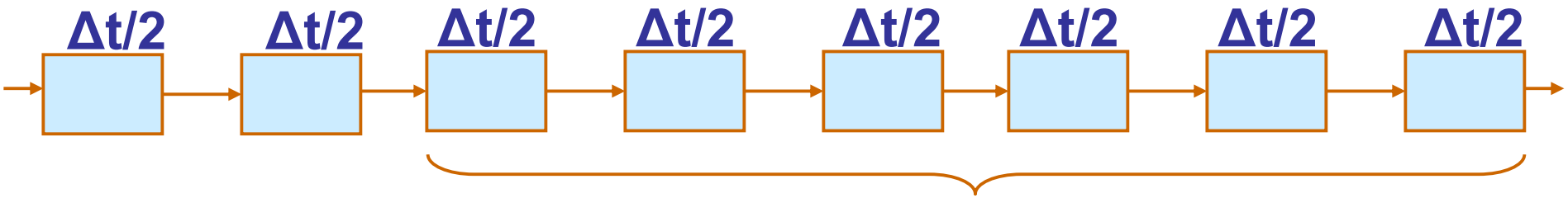
例题解答

7. 超标量超流水线处理机，一个时钟周期分为两个流水级，加法部件和乘法部件都为6个流水线，“取指令”和“指令译码”仍分别为一个流水级，每个流水级能同时发射一条加法和一条乘法指令，结构如下图。综合5）和6）的分析，我们可以知道，计算所需要的最短时间为：

$$T=[2+6+(512-1)+6]\times 10ns=5250ns$$



乘法功能部件为6个流水线



加法功能部件为6个流水线

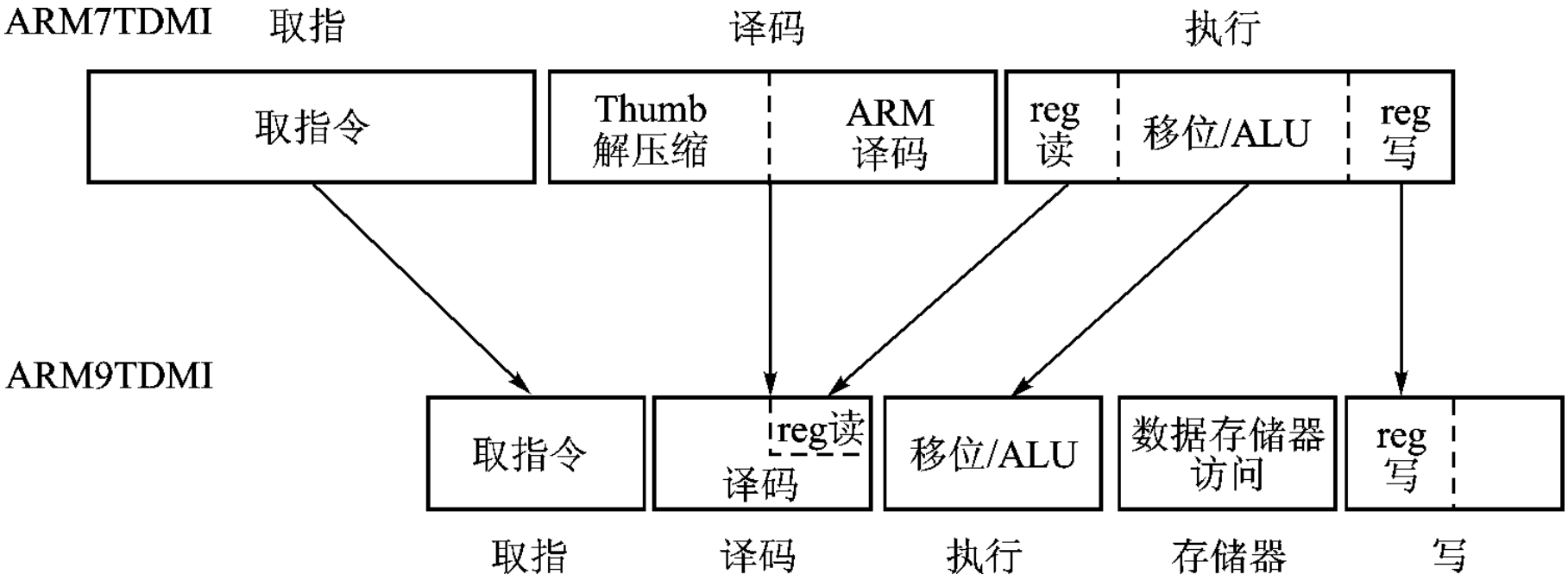
5.5 ARM流水线处理器举例

ARM架构的处理器内核有ARM7TDMI、ARM9TDMI、ARM10TDMI、ARM11TDMI及Cortex等。

ARM7TDMI处理器核采用了3级流水线结构，指令执行分为取指、译码和执行等3个阶段。

ARM9TDMI处理器内核采用了5级流水线

ARM处理器内核


ARM7TDMI
取指
译码
执行
取指令
Thumb
解压缩
ARM
译码
reg
读
移位/ALU
reg
写
ARM9TDMI
取指令
reg读
译码
移位/ALU
数据存储器
访问
reg
写
取指
译码
执行
存储器
写

5.5 ARM流水线处理器举例

主要把3级流水线中的执行阶段的操作进行再分配，即把执行阶段中的“寄存器读”插在译码阶段中完成；把“寄存器写”安排另一级(即第5级完成)，同时，在该级之前，再安排了1级(存储器访问)。因此，ARM9TDMI与ARM7TDMI相比，取指必须快1倍，以便在译码阶段，同时可执行“寄存器读”操作。

ARM9TDMI处理器内核的另一个显著特点是采用指令和数据分离访问的方式，即采用了指令快存I-Cache和数据快存D-Cache。这样，ARM9TDMI可以把数据访问单独安排1级流水线。

5.5 ARM流水线处理器举例

ARM10TDMI在同样的工艺，同样的芯片面积，ARM9TDMI的性能2倍于ARM7TDMI；而ARM10TDMI也同样2倍于ARM9TDMI。ARM10TDMI在系统结构上主要采用提高时钟速率和减少每条指令平均时钟数据CPI两大措施。

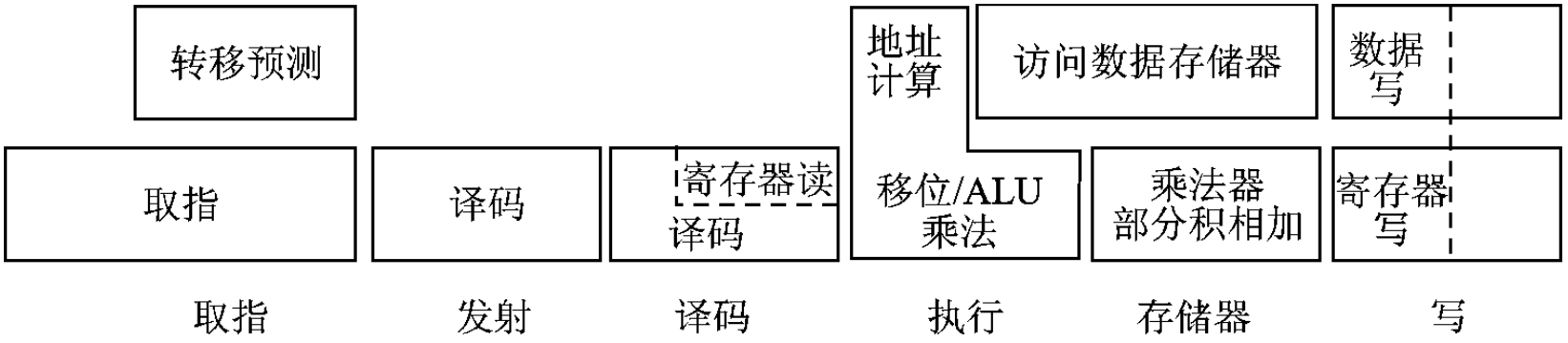
1.提高时钟速率

ARM9TDMI的5级流水线中的4级负担已很满了，当然可以扩充流水线的级数来解决。但是，由于“超级流水线”结构较复杂，因此，只有在比较复杂的机器才采用。ARM10TDMI仍保留与ARM9TDMI类似的流水线，而通过提高时钟速率来优化每级流水线的操作。

5.5 ARM流水线处理器举例

下图是ARM10TDMI采用6级流水线的示意图，与ARM9TDMI的5级流水线相比，ARM10TDMI只需比ARM9TDMI稍快一些的存储器来支持6级流水线。插入了新的一级流水线，允许更多时间去指令译码；只有当非预测转移执行时，才会损害该流水线性能。由于新一级流水线是在寄存读之前插入，它没有新的操作数依赖，所以也不需要新的前进路径。该流水线通过转移预测机构和提高时钟速率，仍可得到与ARM9TDMI差不多的CPI。

ARM10TDMI采用6级流水线



2.减少CPI

上述增强流水线的措施把时钟速率提高50%，可以不损害CPI；若把时钟速率进一步提高的话，就会影响CPI。因此，要采取新的措施来减少CPI。

ARM7TDMI由于采用单一32位存储器，因而存储器几乎占用每一个时钟周期，ARM9TDMI采用指令与数据分离的存储器，虽然数据存储器只有50%的负载，而指令存储器仍几乎占用每一时钟周期。很明显要改进CPI，必须以某种方式来增加指令存储器的带宽。ARM10TDMI采用64位存储器的结构来解决上述的指令存储器的瓶颈问题。

5.5 ARM流水线处理器举例

- **Cortex处理器采用ARMv7体系结构。基于ARMv7体系结构的ARM处理器已经不再延用ARM加数字编号的命名方式,而是以Cortex命名。基于v7A的称为“Cortex-A系列”,基于v7R的称为“Cortex-R系列”,基于v7M的称为“Cortex-M系列”。 Cortex-A系列是针对日益增长的，运行包括Android、Linux、Windows CE和Symbian操作系统在内的消费娱乐和无线产品；Cortex-R系列是针对需要运行实时操作系统来进行控制应用的系统，包括汽车电子、网络和影像系统；Cortex-M系列则是为那些对开发费用非常敏感同时对性能要求不断增加的微控制器应用所设计的。**

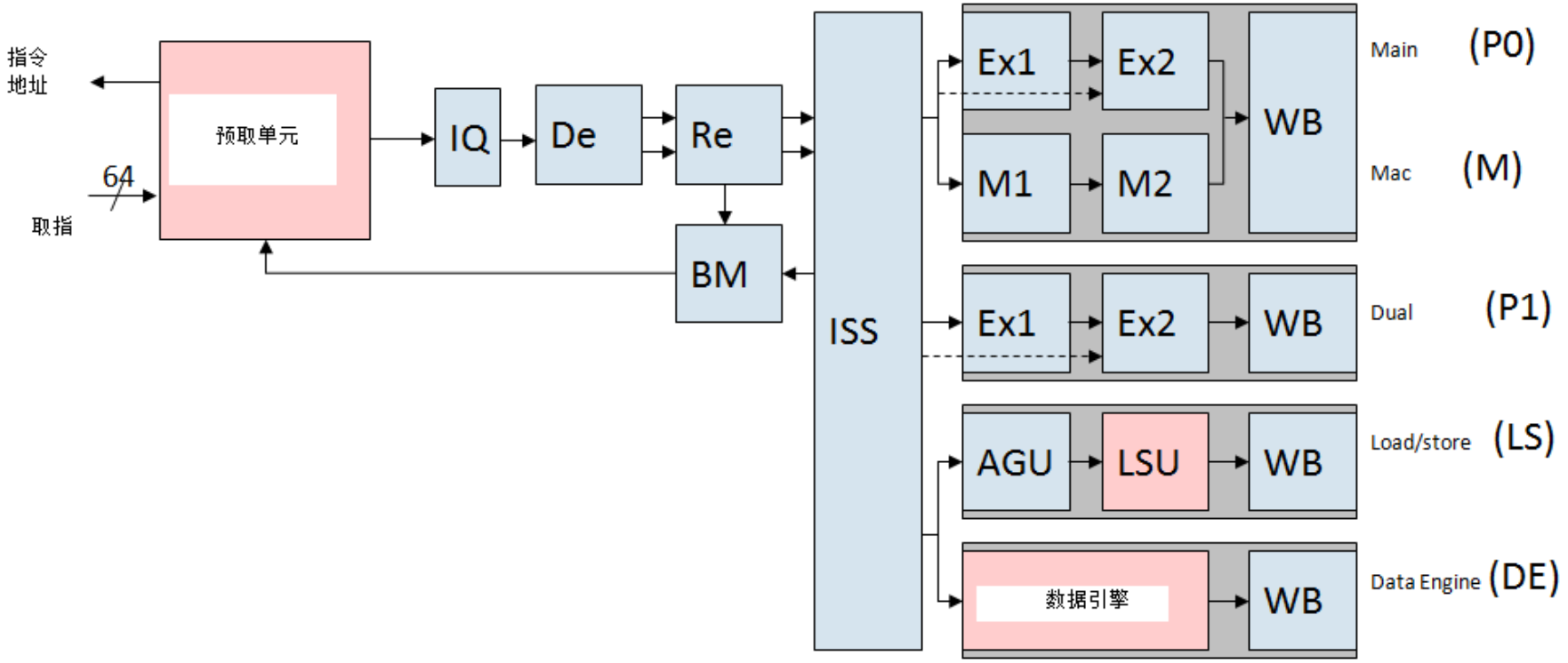
5.5 ARM流水线处理器举例

- **Cortex-A9 处理器为ARM v7-A 体系结构，支持 ARM、Thumb-2和ThumbEE，具有安全扩展（TrustZone）、可选Jazelle DBX、可选媒体处理引擎(NEON + FPU-D32)、可选浮点单元(FPU-D16)，支持MMU实现VMSA v7。它是高性能整型内核，双发射、乘法后端流水线、乱序指令执行、可配置大小的I和D Cache。调试和跟踪只跟踪程序流指令。Cortex-A9流水线如图5-56所示。Cortex-A9流水线的各功能段如下：**

5.5 ARM流水线处理器举例

- **IQ:　　指令排队（Instruction Queue）**
- **Re:　　寄存器重命名（Register renaming）**
- **BM:　　分支监控（Branch Monitor）**
- **P0:　　主执行流水线（Main execution pipeline）**
- **M: MAC 流水线**
- **P1:　　第二 (“dual”) 执行流水线**
- **AGU:　　地址产生单元（Address Generation Unit）**
- **LSU:取/存单元（Load/Store Unit）**
- **DE:　　数据引擎（Data Engine） - (NEON 和/或FPU)流水线**

5.5 ARM流水线处理器举例



本章重点

- 重叠解释方式
- 流水线的分类
- 流水线处理机的主要性能（吞吐率、加速比、效率）
- 流水线的时空图、流水线瓶颈段的处理
- 流水机器的相关处理
- 非线性流水线的调度

本章重点

- 向量的流水处理
- 向量流水处理机
- 向量指令之间的链接技术
- 单发射，多发射
- 超标量处理机的指令执行时序及性能
- 超流水线处理机的指令执行时序及性能
- 超标量超流水线处理机的指令执行时序及性能

第5章 作业5

- 5-1
- 5-3
- 5-4
- 5-5
- 5-6
- 5-12