

微机原理及接口技术

Hardware Principles and Interfacing of Modern Computer

Lecture 3: Microprocessor and It's Architecture

陈启军, 张伟

Email: zhang_wi@mail.tongji.edu.cn

Dept. Of Automation, TongJi University



Content

- **基本概念: 微处理器 MPU / CPU**
 - 基本原理: 指令控制
- **基本概念: 算逻单元 ALU**
 - 全加器
 - 定点与浮点部件
 - 完整的运算器
- **基本概念: 寄存器**
- **改进技术**
 - 改进1: 指令流水线 Instruction Pipeline
 - 改进2: 超标量技术 SuperScalar
 - 改进3: 进一步的思考

Microprocessor

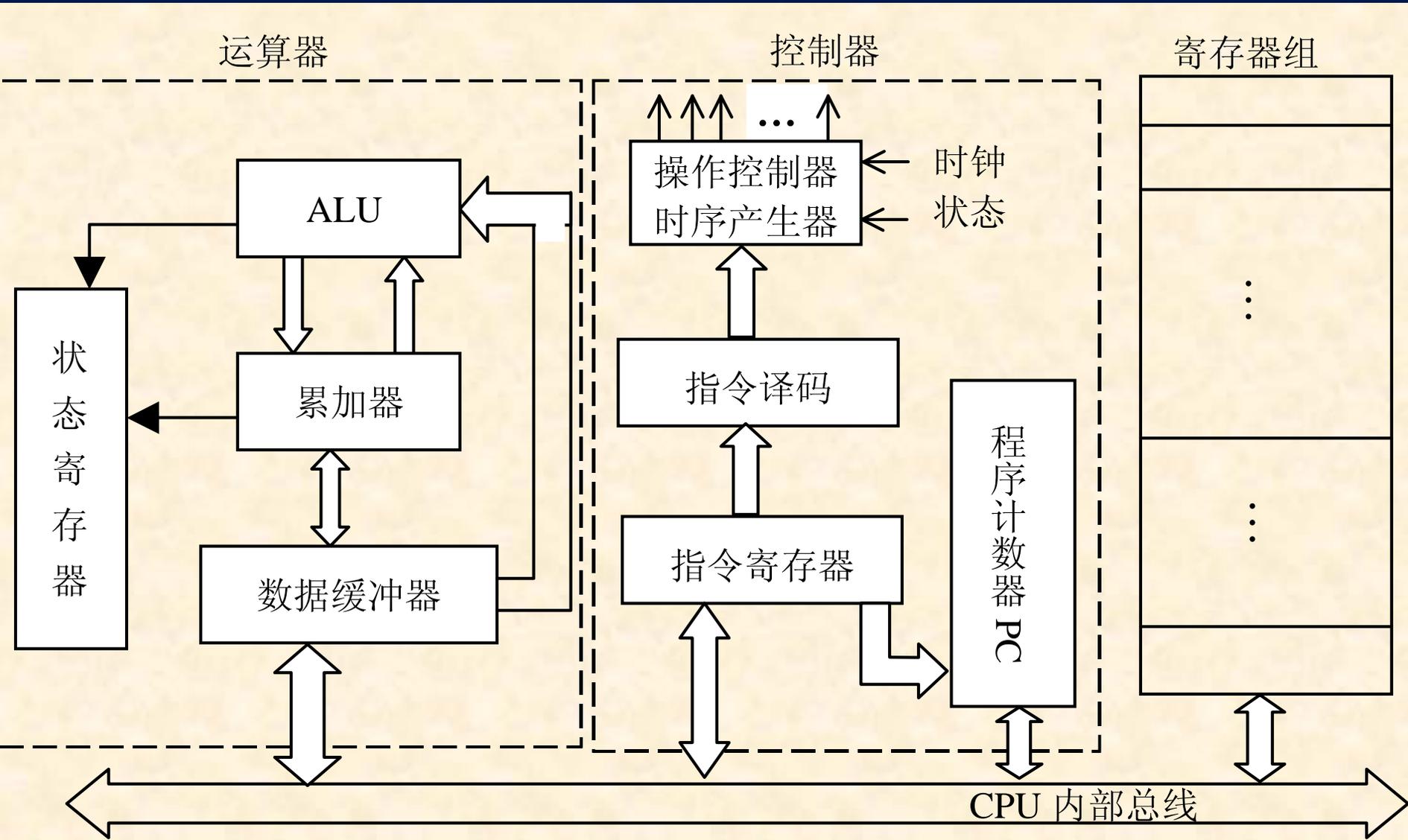
● 基本概念：微处理器(MPU / CPU)

- 微处理器也称中央处理器（Central Processing Unit, CPU）。它是由控制单元（Control Unit）、算术／逻辑运算单元（Arithmetic Logic Unit, ALU）和存储单元（Memory Unit，包括寄存器和高速缓存）三大部分组成的。
- 三部分通过互连机构的连接，互相配合，共同完成对指令信息和数据信息的分析、判断、运算，从而控制计算机各部件协调工作。

Microprocessor

● 基本原理：指令控制

- 所谓指令控制，就是指计算机严格按照程序的顺序执行。一条指令往往是由若干个微操作信号的组合来实现的，CPU从内存取出指令，译码后产生微操作信号，然后把各种微操作信号送往相应的部件，从而控制这些部件按指令的要求开始动作，这就是操作控制功能。
- 回溯：图灵机, 分析机, V. Neumann机



CPU主要组成部分逻辑框图

Arithmetic Logic Unit

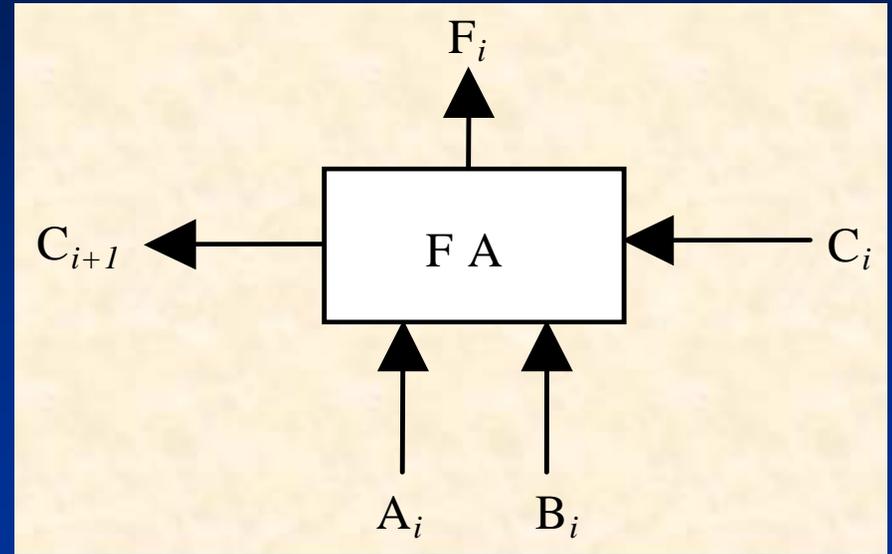
● 基本概念:算逻单元(ALU)

- 算逻单元全称为算术/逻辑运算单元(Arithmetic Logic Unit)。主要用来完成算术运算和逻辑运算。在实现时，一般采用全加器和先行进位逻辑电路进行组合，从而能实现高速运算功能。
- 一位全加器 (FA) 的逻辑表达式为：

$$F_i = A_i \oplus B_i \oplus C_i$$
$$C_{i+1} = A_i B_i \oplus (A_i \oplus B_i) C_i$$

Arithmetic Logic Unit

- 构成ALU的最基本构件：
一位全加器(Full Adder)
- 思考：如何由一位全加器
构成多位全加器？(级联问
题)？如何提高运算效率？
(硬件加速问题)

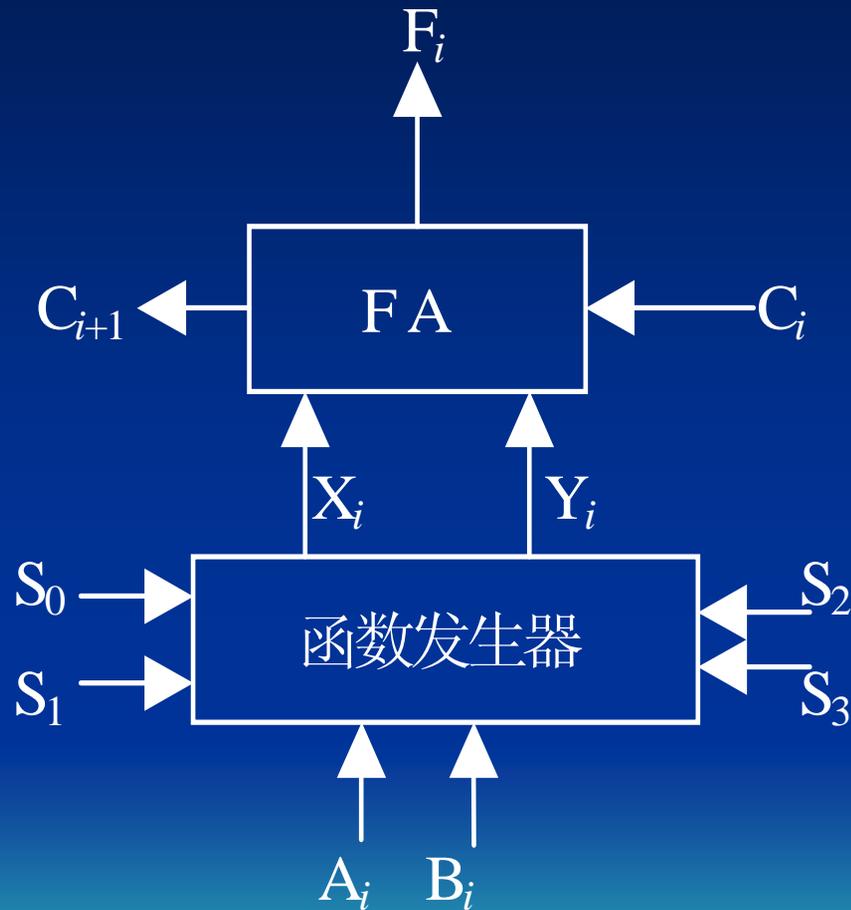


Arithmetic Logic Unit

● 全加器的功能扩展:

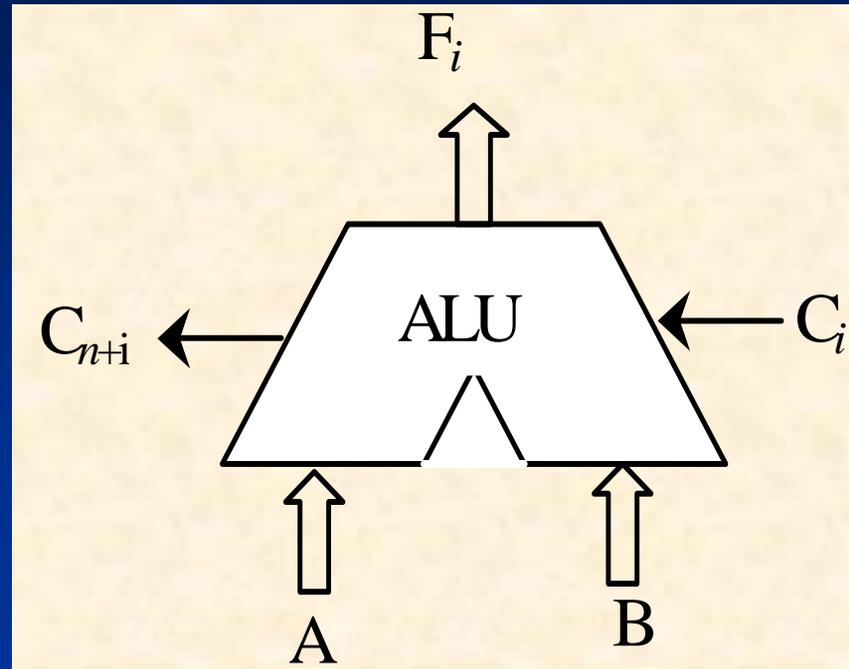
上述全加器只能实现算术运算，为了使它具有逻辑运算功能，必须对其功能进行扩展。扩展时通过增加一个函数发生器，在4个控制开关 S_0 、 S_1 、 S_2 、 S_3 的控制下，使 A_i 、 B_i 经函数发生器产生新的组合函数 X_i 、 Y_i ，然后再将 X_i 、 Y_i 和进位 C_i 通过全加器进行全加。这样，不同的控制开关组合，就可以得到不同的组合函数，从而能够实现多种算术和逻辑运算，如图2.3所示。ALU的逻辑符号如图2.4所示。

Arithmetic Logic Unit



一位全功能全加器

Arithmetic Logic Unit

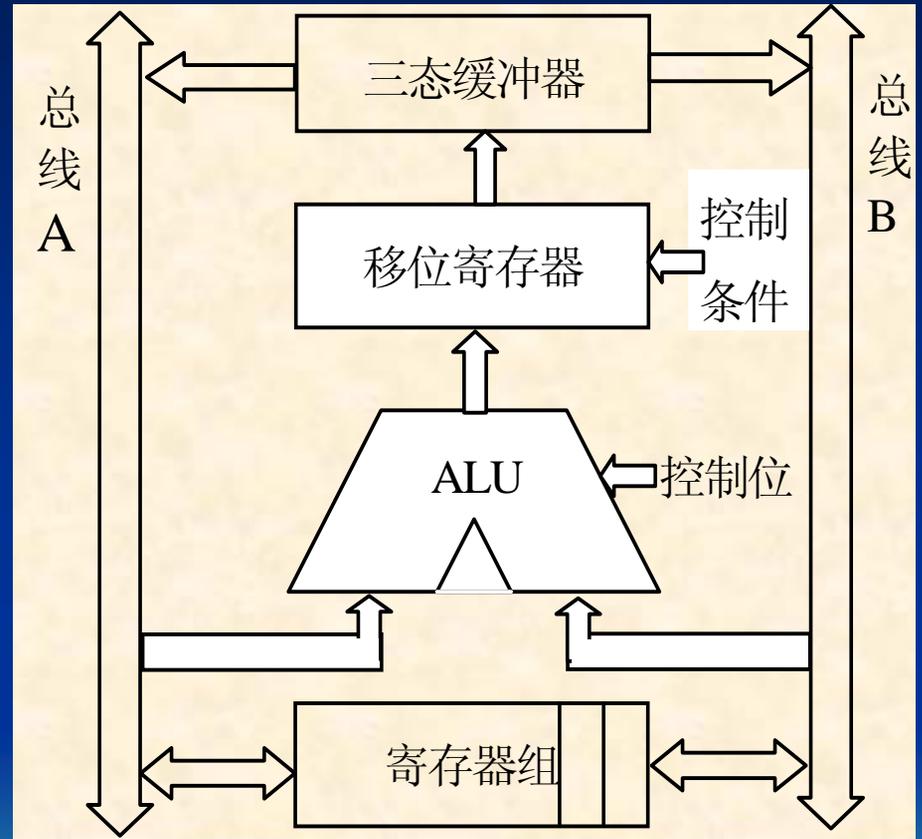


由一位的全功能全加器经过复合/级联/优化设计后得到实际中的ALU

Arithmetic Logic Unit

双总线定点运算运算器

- 在ALU的基础上加上与暂存/传输有关的各种寄存器,总线接口部件等电路就得到了完整的运算器



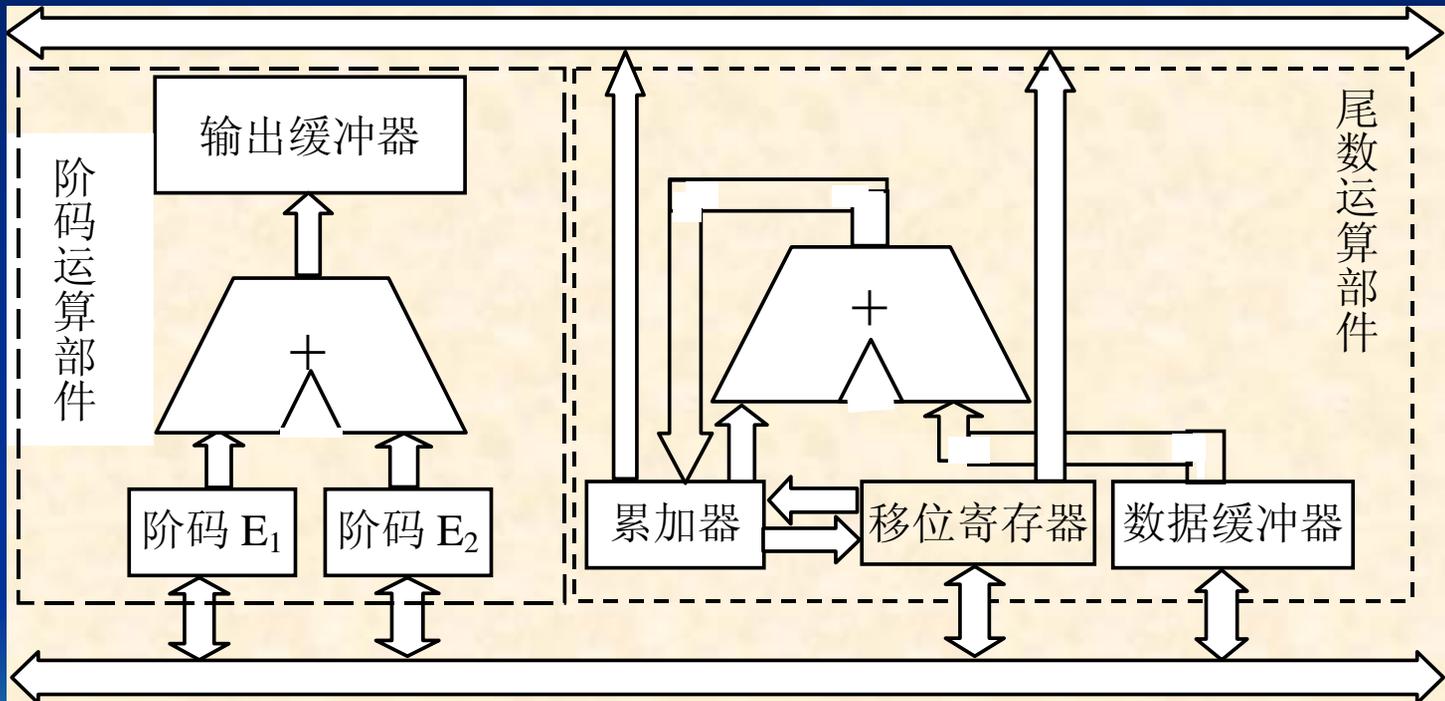
Arithmetic Logic Unit

● 定点与浮点部件

- 定点数部件: 定点数是小数点位置固定不变的数。由于小数点位置固定, 小数点就不再使用记号“.”来表示。我们通常将定点数分为定点小数和定点整数。
- 浮点部件: 浮点数的小数点位置不固定, 在表示一个浮点数 N 时, 一般记为 $N=RE \times M$ 的形式。其中 M 称为浮点数的尾数, 是一个纯小数, E 为阶码, 是一个纯整数, 基数 R 可取2、8或16, 一般取2。为了在机器中表示一个浮点数, 通常要将寄存器分成两部分, 一部分存放阶码 E , 另一部分存放尾数 M 。

Arithmetic Logic Unit

● 浮点运算器的一般结构



Registers

● 基本概念：寄存器

- 寄存器是沟通硬件电路与软件代码之间的桥梁。软件通过读写寄存器获取硬件的状态并实现对硬件的操纵。
- 现代CPU中通常包含了大量的寄存器，既有通用用途，又有专门用途

● 用户可见寄存器

- 用户可见寄存器，就是指用户使用机器语言、汇编语言或其他高级语言可以直接访问和控制的寄存器。
- 如：各个通用寄存器

Registers

● 对用户透明寄存器

- 这类寄存器对用户来说是不可见的。它是在操作系统的作用下，在CPU中起着控制计算机操作的作用。对用户透明寄存器至少包括程序计数器（PC）、指令寄存器（IR）等。

● 状态寄存器

- 状态寄存器由一个或一组寄存器构成。用来保存由算术指令和逻辑指令运行或测试后的结果、建立的各种条件码内容以及系统中断和系统工作状态等信息。

Improvement

● 改进1: 指令流水线(Instruction Pipeline)

● 基本原理: 指令流水线

流水线处理机把一条指令的执行分成 K 个步骤，或称级(Stages)。每一级能够在—个时钟周期内完成。在每个时钟周期里，处理机启动并执行—条指令。也就是说处理机的流水线共有 K 级，同时可重叠执行的指令总条数为 K ，并且，每条指令处在不同的执行阶段。而这种级的划分是和指令相应的执行部件有关，它必须保证在不同的执行时间内，操作所占用的部件是不相同的，这样才不会出现资源竞争的现象，才能保证流水线的正常运行。

Improvement

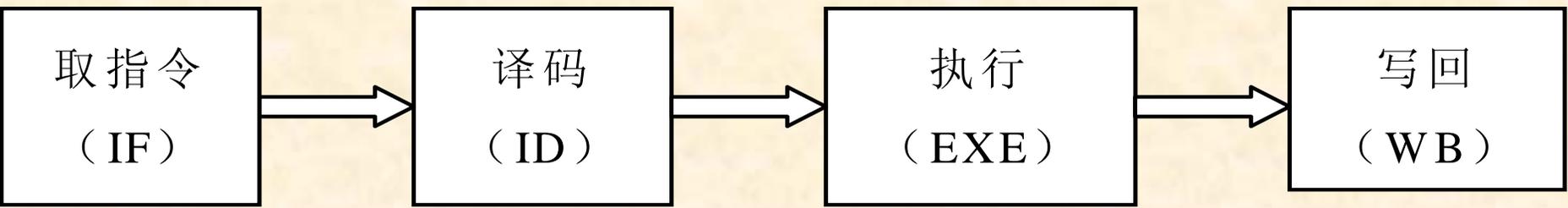
例如，一条典型指令的执行划分为4个步骤：

(1) 取指令 (Instruction Fetch) 阶段，用IF表示。它是指将一条指令读到指令缓冲器的过程；

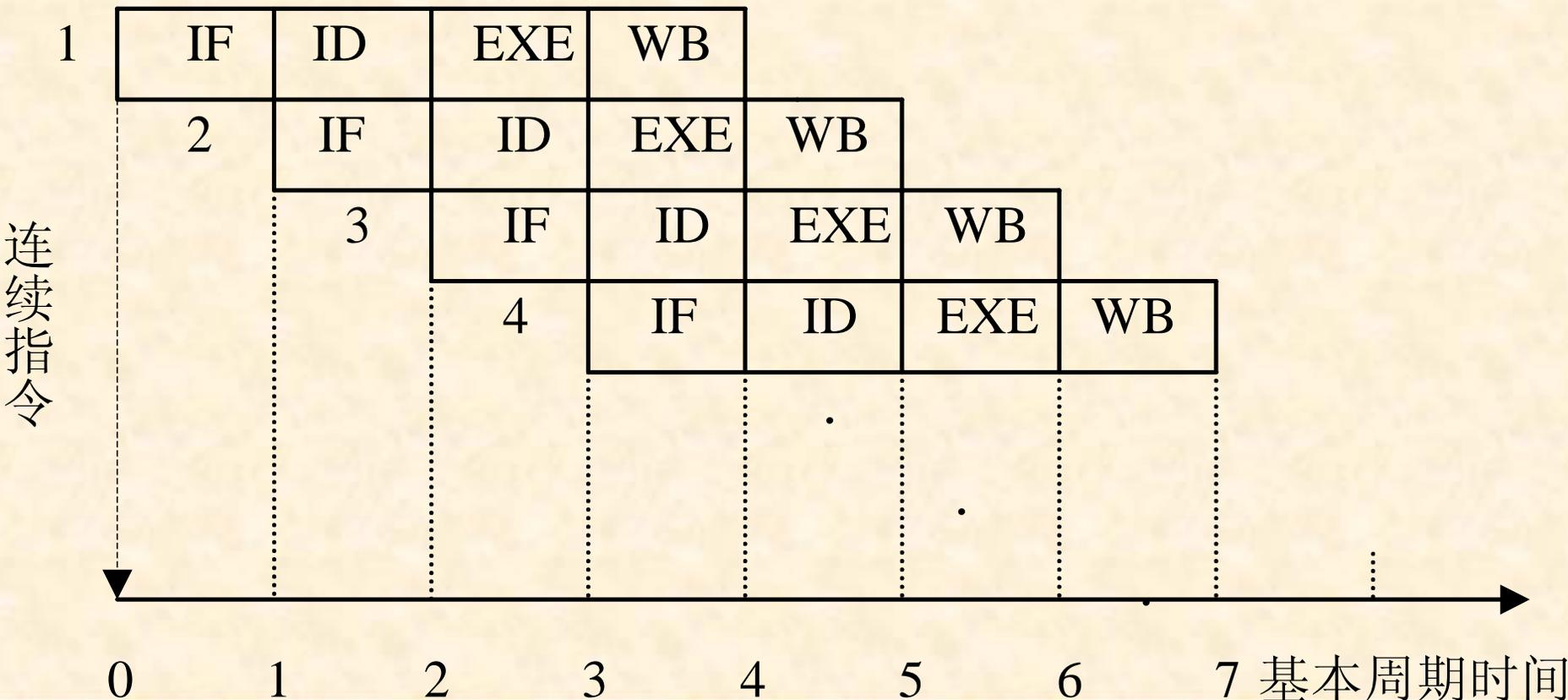
(2) 译码 (Instruction Decode) 阶段，用ID表示。它包括指令译码和从寄存器中读操作数 (若有操作数的话) 两种操作；

(3) 执行 (Execute) 阶段，用EXE表示。在这个阶段中完成特定的操作。若在指令中有指定存放目标操作数的地址，则将结果写入此位置；

(4) 写回 (Write Back) 阶段，用WB表示。用来将最终结果写入相关的寄存器或存储器中。



(a) 四阶段流水线



(b) 四阶段流水线时间重叠情况
指令流水线思想

Improvement

🌐 相关的基本概念:

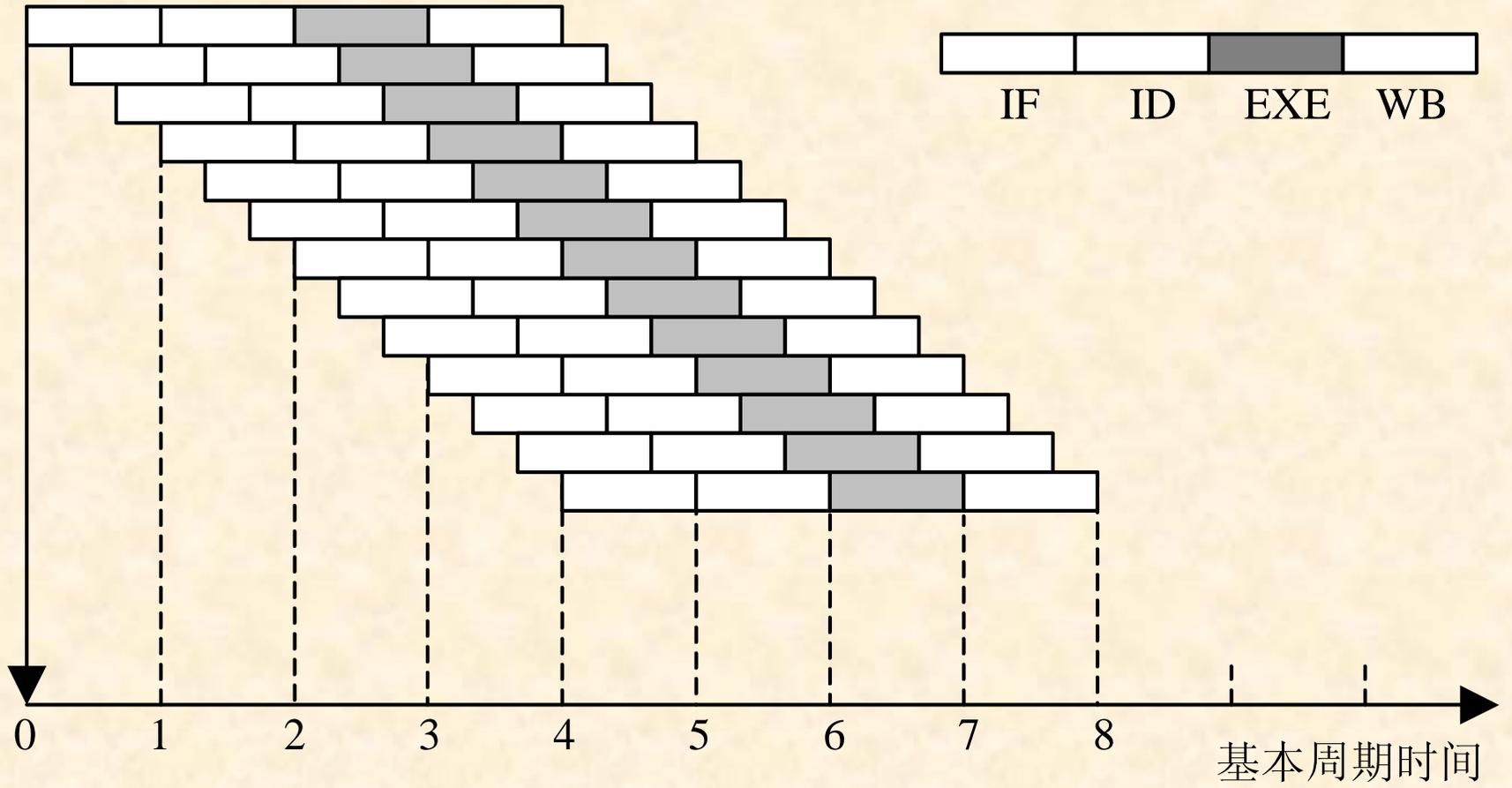
- 指令流水线周期，是指驱动流水线操作的时钟周期。通常它等于在流水线中完成单级操作所需要的最大时延，有时也称处理器周期。
- 指令发射时延，指发射两条相邻指令间的处理器周期数。
- CPI (Cycles Per Instruction)，每条指令经过整个流水线所需周期数，它与各指令有关，不同的指令，CPI也不相同。因此，一般使用平均CPI表示。
- 指令级并行度ILP (Instruction Level Parallelism)，是指当指令流水线被充分利用时，一个基本周期内并行执行的指令数。这里的基本周期等效于常规的标量流水机的时钟周期。
- 指令发射速率，指在每个周期内发射的指令数，在数值上等于超标量处理机的级数。

Improvement

思考: 流水线的性能估计和影响因素分析

流水线的设置受到很多条件的限制。流水线的级数越多, 数据相关 (如一条指令还没有完成对某寄存器的写操作, 接下来的一条指令就要用这个寄存器的内容, 就发生必须等待上条指令完成写操作, 造成流水线效率下降)、分支程序造成的控制相关 (下一条指令可能不是要执行的指令) 等副作用越大, 它们会大大影响流水线的效率。

连续指令

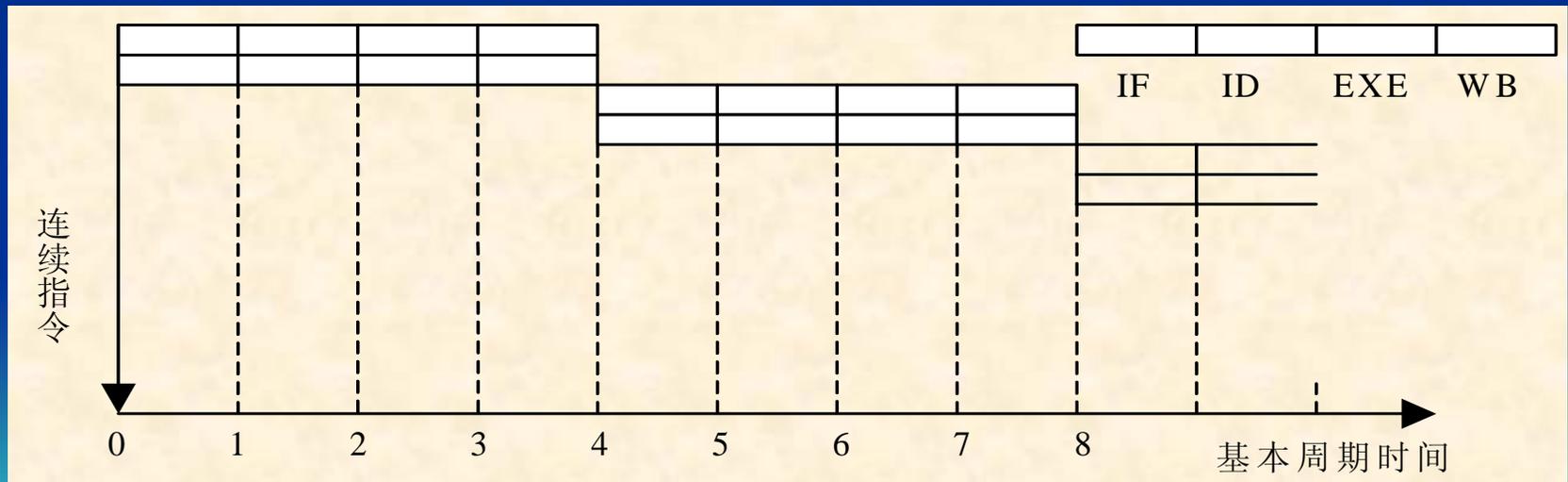


$n=3$ 的单发射超级流水线

Improvement

改进2: 超标量技术

超标量 (superscalar) 处理机采用资源重复的并行性思想, 设置多条指令流水线和多个功能部件。每个周期发送多条指令 (即指令发射速率大于1), 同时并行地对多条指令进行流水处理。



$m=2$ 的单纯超标量技术

Improvement

● 思考: 能不能继续改进

- 回顾三大基本思路: 资源共享, 时间复用和空间复用
- 流水技术 + 超标量技术
- 超长指令字技术
- RISC技术诞生以及今天CISC和RISC的走向融合
 - Register Window技术
- 超线程技术
- **You Think It !**

Example: 8086/8088

● Intel 8086和Intel 8088溯源

● Intel 8086/8088的基本技术指标

- 主频：4.77MHz
- 16位的内部体系结构
- 数据线16位/8位
- 20位地址线
- 16位I/O端口地址线
- 其他

Example: 8086/8088

🌐 微处理器的制造工艺限制

- 引脚数限制
- 芯片面积限制
- 器件速度限制

🌐 16位微处理器的基本特点

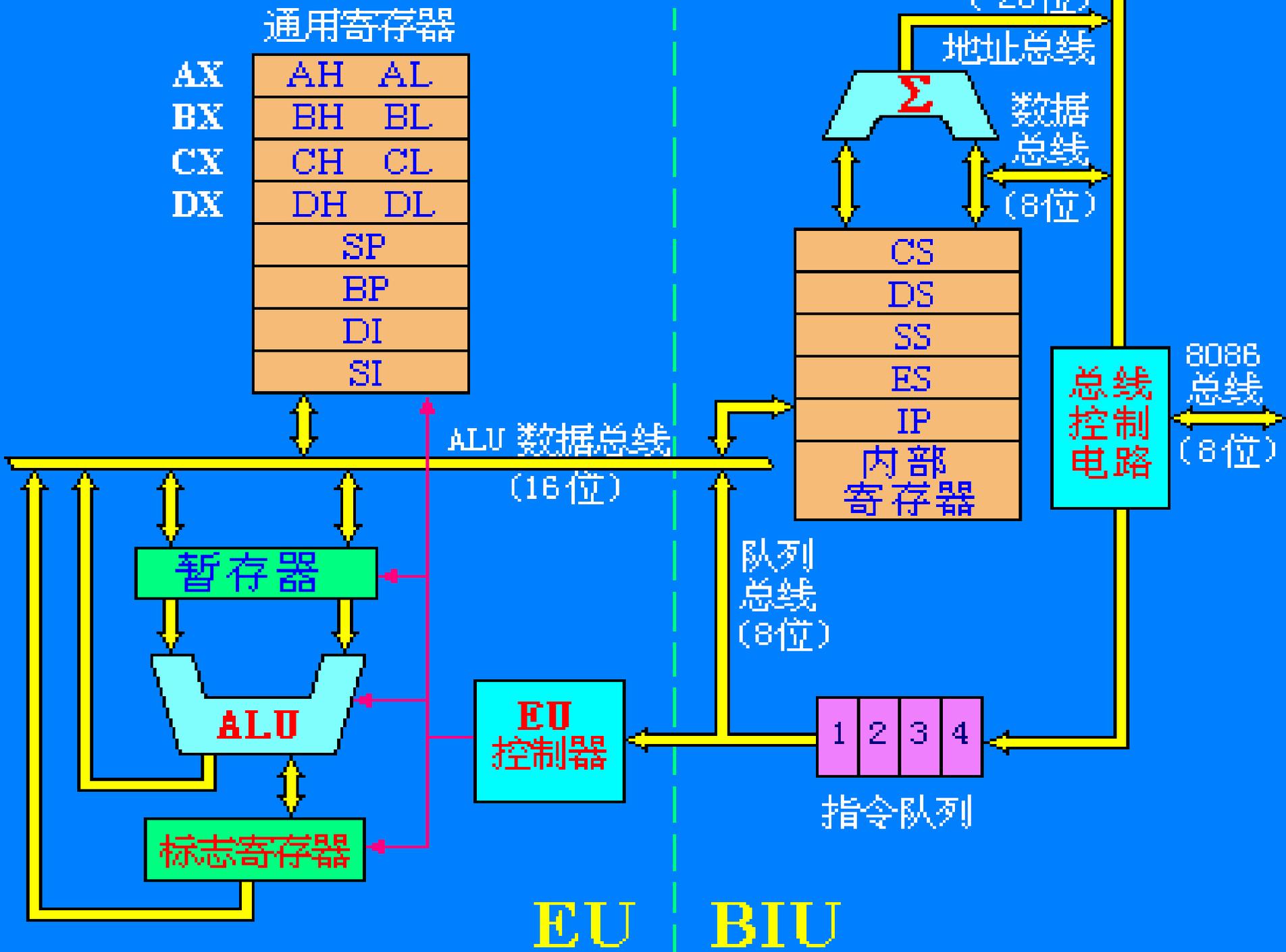
- 引脚功能复用
- 单总线累加器结构。
- 可控三态电路
- 总线分时复用

Example: 8086/8088

8086CPU的内部结构

由两大部件组成

- 总线接口部件BIU (Bus Interface Unit)
- 执行部件EU (Execution Unit)



Example: 8086/8088

● 总线接口部件BIU

● 构成

- 四个段寄存器
- 20位地址加法器
- 指令队列
 - 8088芯片：4字节
 - 8086芯片：6字节
- 总线控制逻辑
- 16位指令指针

Example: 8086/8088

● 执行部件EU

- 负责指令译码、执行与数据处理

● 构成

- EU控制逻辑
- 暂存器
- 算术逻辑运算单元ALU
- 寄存器组
- 标志寄存器

Example: 8086/8088

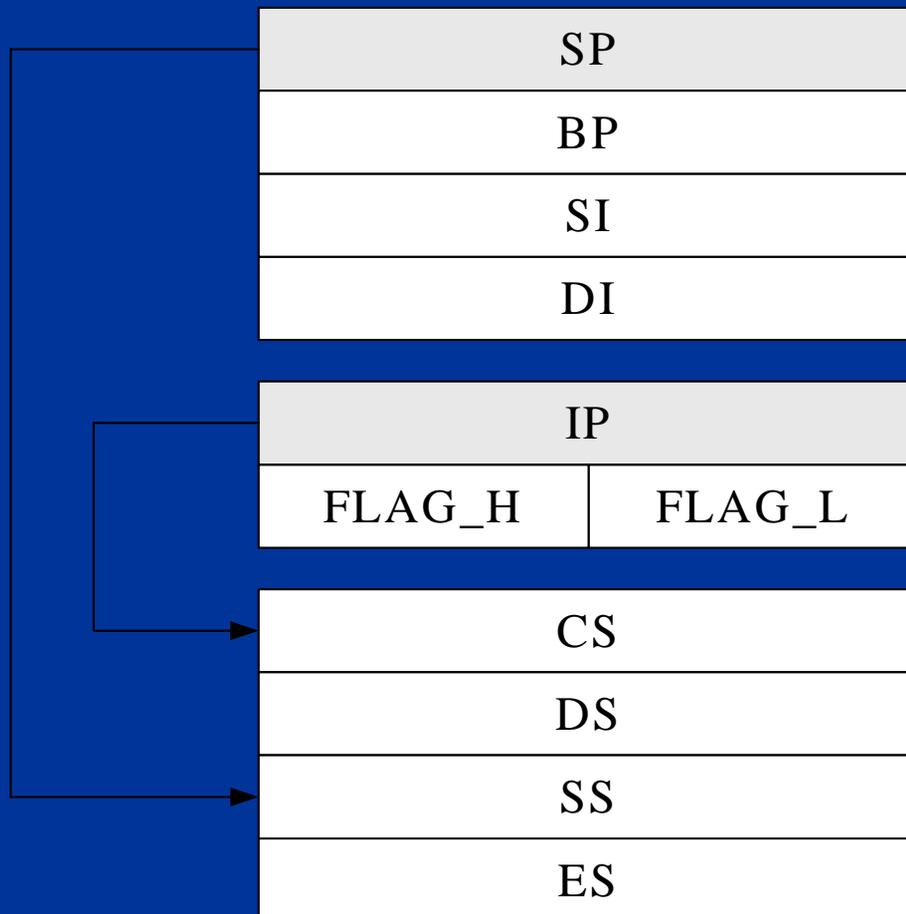
8086/8088的寄存器结构

- 通用寄存器组—AX, BX, CX, DX
- 指针和变址寄存器
 - SP ——(Stack Pointer)堆栈指示器
 - BP ——(Stack Pointer)基址指示器
 - SI ——(Source Index Pointer)源变址指示器
 - DI ——(Destination Index Pointer)目标变址指示器
- 段寄存器—CS, DS, SS, ES
- 指令指针寄存器—IP
- 标志寄存器F (又称PSW)
 - 状态标志: SF, ZF, CF, AF, PF, OF
 - 控制标志: DF, IF, TF

8086 内部寄存器的组成

AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL

(A) 累加器
(HL) 基数
(BC) 计数
(DE) 数据

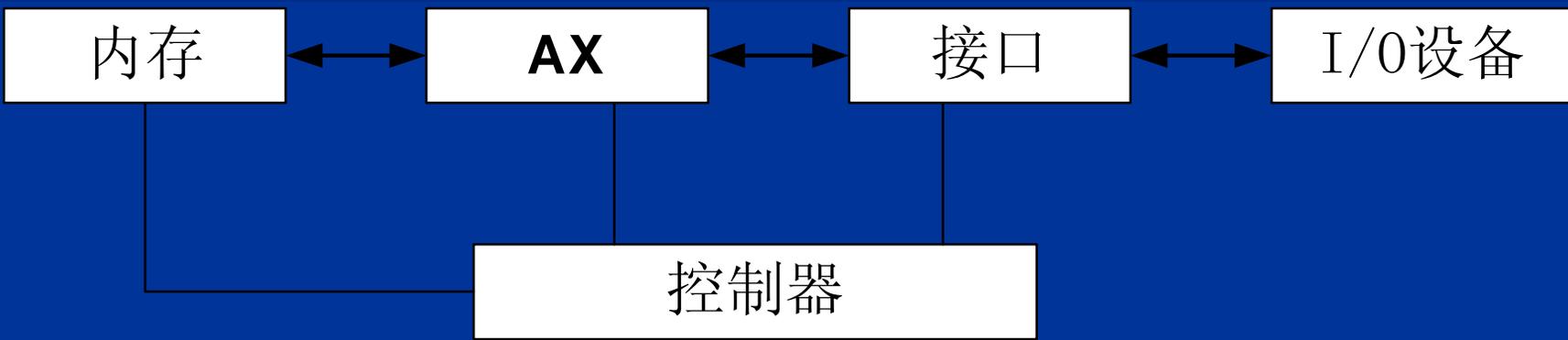


(SP) 堆栈指针
基数指针
源变址寄存器
目的变址寄存器
(PC) 指令指针
(PSW) 状态标志
代码段寄存器
数据段寄存器
堆栈段寄存器
附加段寄存器

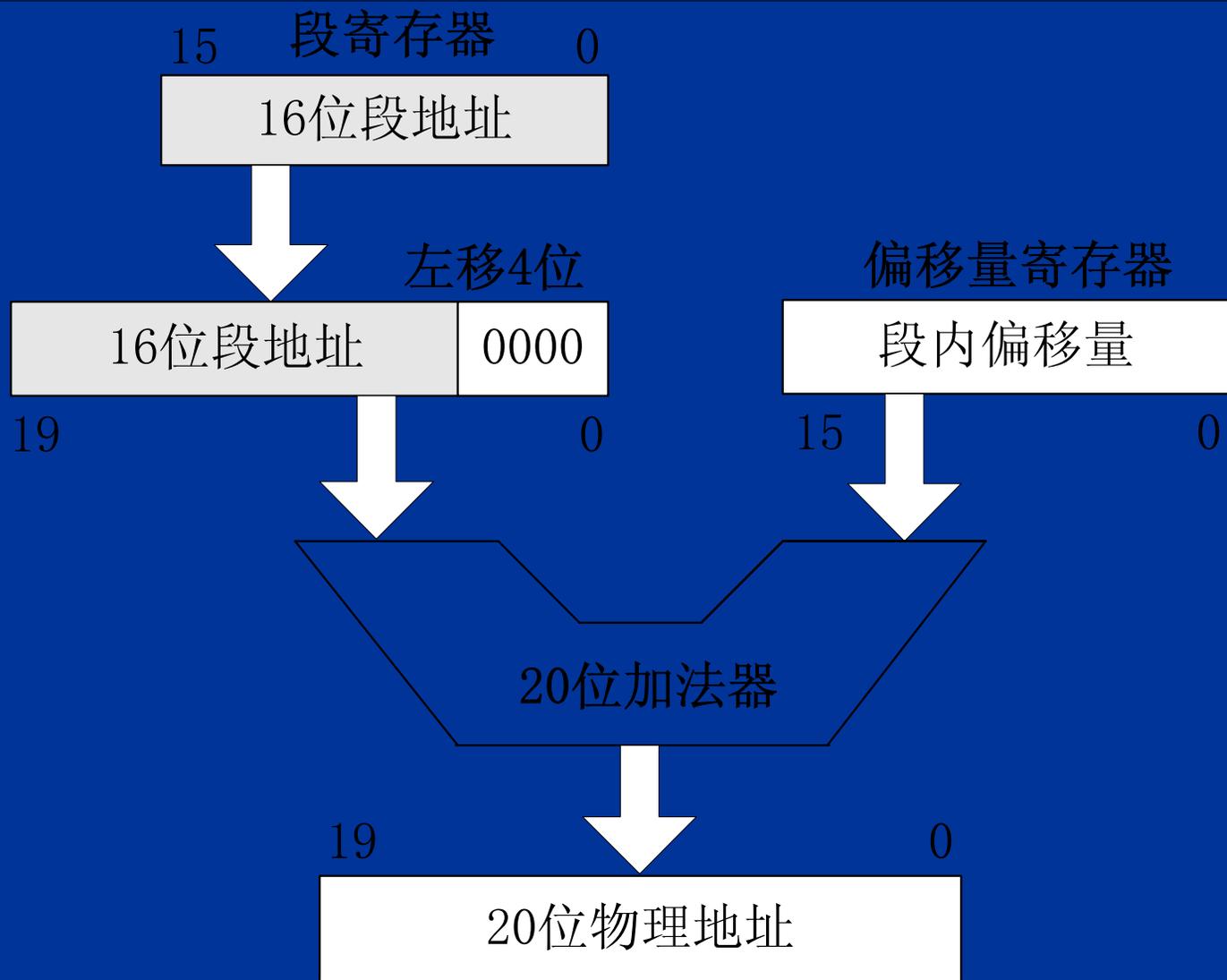
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



标志寄存器FLAGS



主机与外设信息交换示意图



8086/8088物理地址的形成

Example: 8086/8088

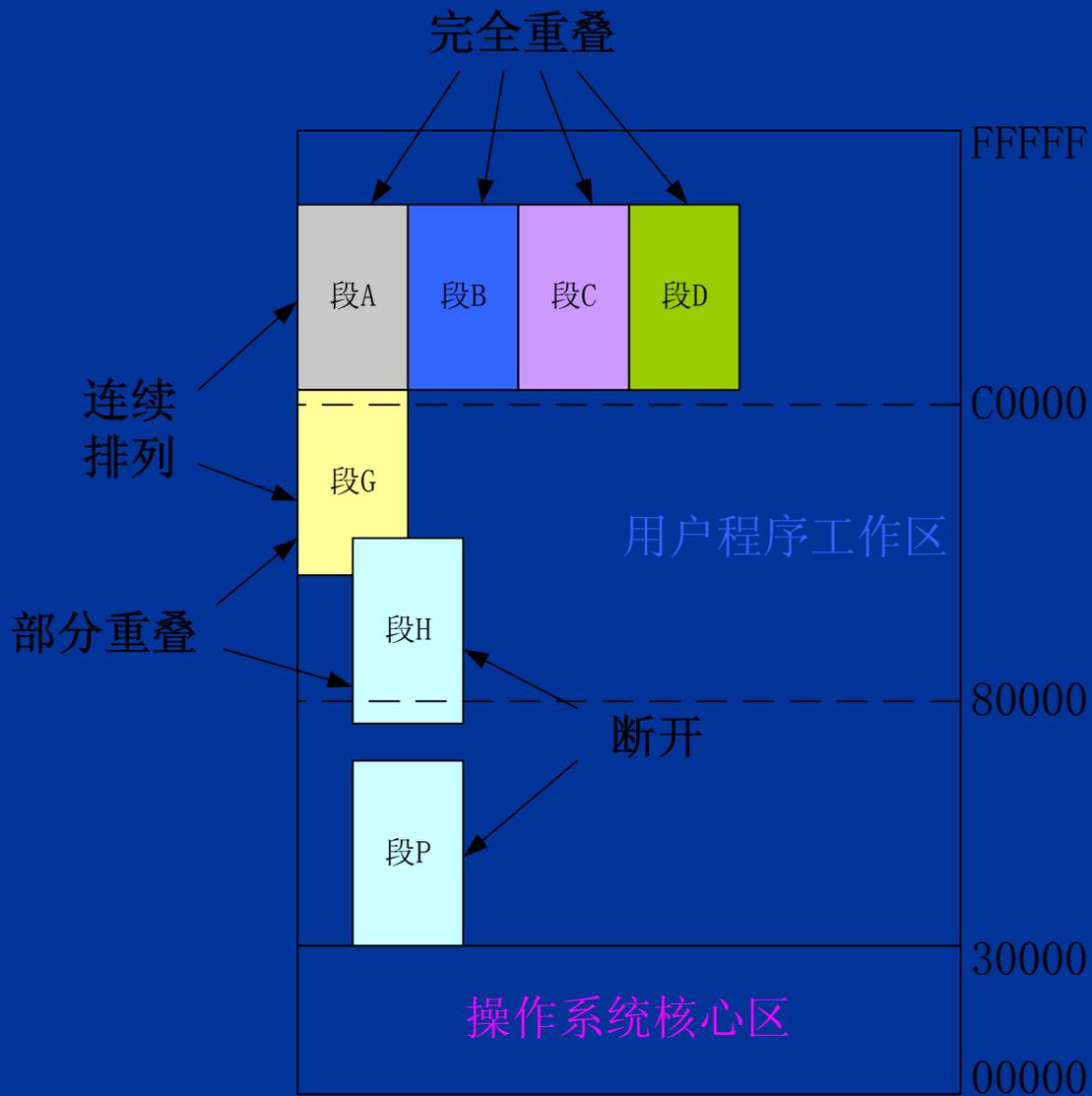
物理地址计算示例

例1:

$$\begin{array}{r} ? \text{ (DS)}=8000\text{H} \\ ? \text{ (BX)}=\text{FFFFH} \\ \hline \text{+)} \quad \text{FFFFH} \text{ —— 偏移量 (16位)} \\ \quad \quad 80000\text{H} \text{ —— 数据段基地址} \\ \hline \quad \quad 8\text{FFFFH} \text{ —— 数据物理地址 (20位)} \end{array}$$

例2:

$$\begin{array}{r} ? \text{ (CS)}=4232\text{H} \\ ? \text{ (IP)}=66\text{H} \\ \hline \text{+)} \quad \quad 66\text{H} \text{ —— 偏移量 (16位)} \\ \quad \quad 42320\text{H} \text{ —— 代码段基地址} \\ \hline \quad \quad 42386\text{H} \text{ —— 指令物理地址 (20位)} \end{array}$$



x86微型机的存储器分段结构

Example: 8086/8088

PSW中标志位的符号表示

标志名	标志值=1	标志值=0
OF 溢出（是/否）	OV	NV
DF 方向（减/增量）	DN	UP
IF 中断（允许/关闭）	EI	DI
SF 符号（负/正）	NG	PL
ZF 零（是/否）	ZR	NZ
AF 辅助进位（是/否）	AC	NA
PF 奇偶（偶/奇）	PE	PO
CF 进位（负/正）	CY	NC

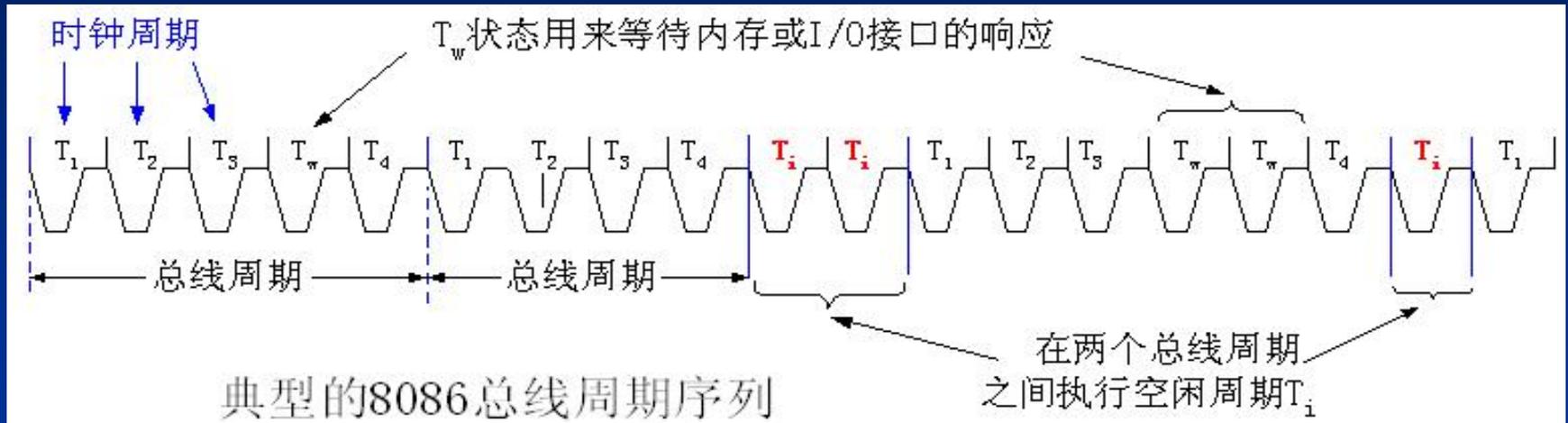
Example: 8086/8088

8086总线周期

- 时钟周期
 - CPU的基本时间计量单位
- 基本总线周期
 - 4个时钟周期组成
- 4个时钟周期分别称为4个状态
 - T1状态, T2状态, T3状态, T4状态
- 参看总线周期时序图

Example: 8086/8088

典型的8086总线周期序列



Example: Pentium

奔腾（Pentium）处理器

- 它是1993年由美国的Intel公司推出。Pentium处理器是复杂指令集计算机（CISC）几十年来设计、努力的结晶，它采用了过去只有大型机和超级计算机才会采用的设计原则，是CISC体系结构的优秀典范。
- Pentium处理器采用了全新的设计，由32位地址线和64位数据总线以及高性能浮点处理部件构成的高性能微处理器。
- 执行单元可以高速完成各种算术和逻辑运算。它包括两个整数ALU和一个浮点ALU，分别执行整数和浮点数的运算。

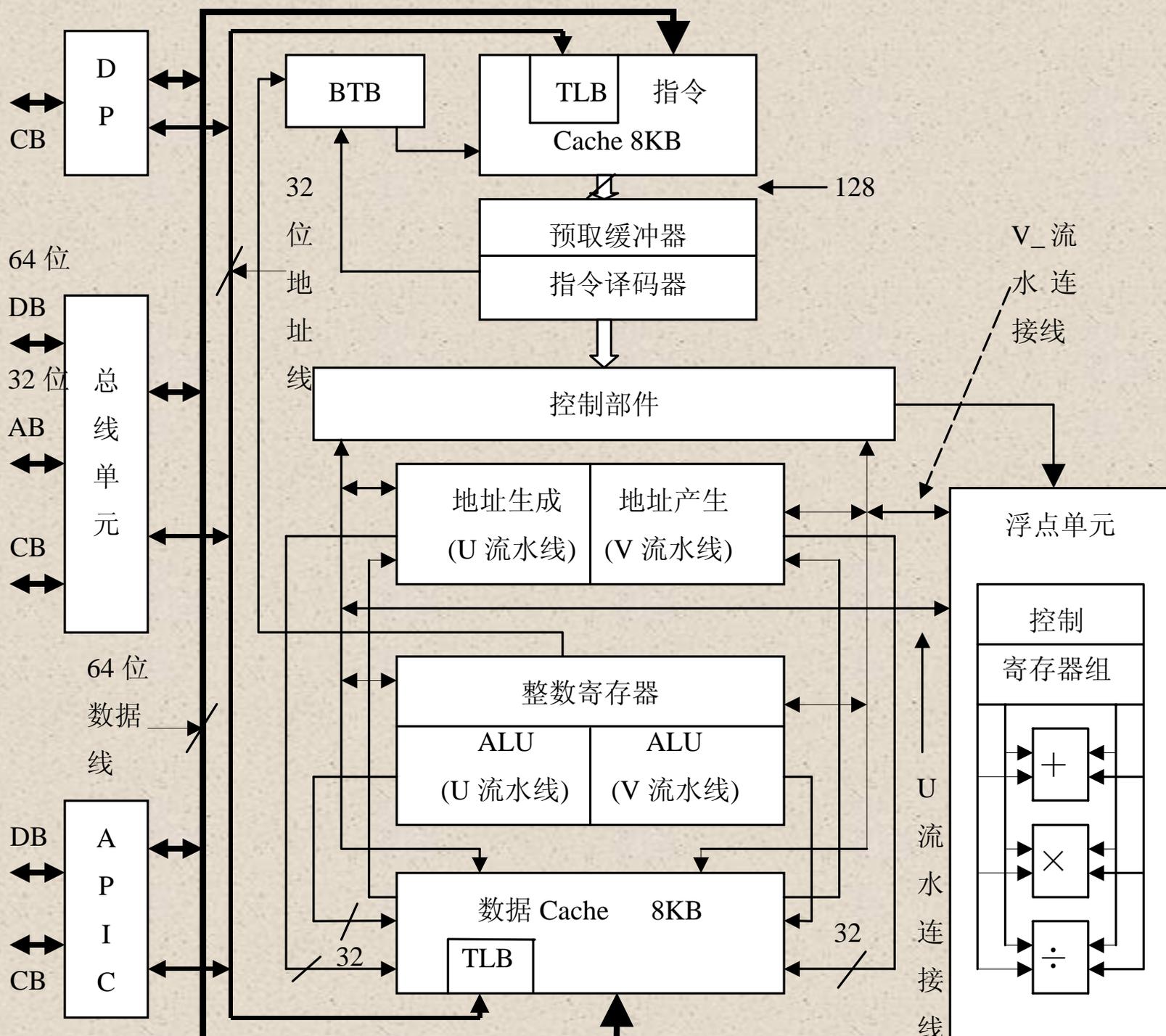


图 Pentium 处理器的内部结构示意图

Example: Pentium

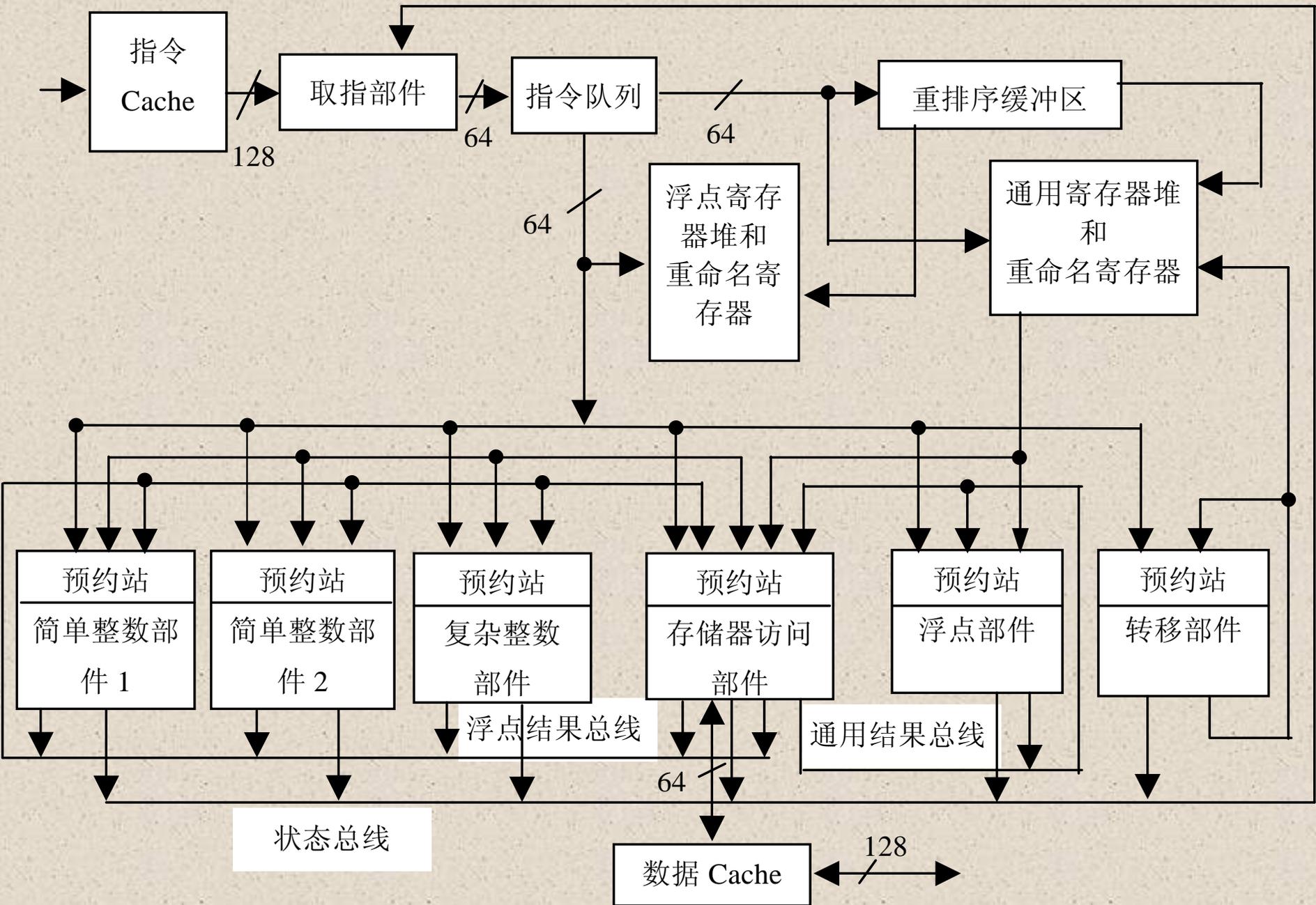
Pentium处理器的特点

- Pentium处理器的结构虽然与80486及以前处理器的结构有很大改进，但是依然保持了与它们之间的兼容性。除此之外，Pentium处理器还具有一些新的特点。
- 指令Cache与数据Cache分开，提高了计算机执行的并行性。
- 采用了超标量技术，在处理器内部设计了两条指令流水线，U流水线和V流水线。
- 浮点部件采用运算流水线结构，使Pentium处理器的速度是i486处理器速度的2~3倍。
- 具有转移指令预测功能。通过转移目标缓冲器（Branch Target Buffer, BTB）来预测转移指令。

Example: Power PC

Power PC处理器

- 它是由IBM、Apple和Motorola三个公司联合开发，采用超标量的RISC体系结构的微处理器。Power PC系列包括601、603、604e、620、630、G₃ (740/750) 和G₄等。它主要是为具有较好软件兼容性的商业应用而设计的。



Power PC 620结构框图
Tongji University

Summary

- 指令控制思想
- ALU的构成和设计
- 寄存器技术
- 流水线技术