

1.6 系统结构中的并行性

1.6.1 并行性概念

1. 并行性的含义与并行性级别

并行性包含同时性和并发性二重含义。

同时性——两个或多个事件在同一时刻发生。

并发性——两个或多个事件在同一时间间隔内发生。



1.6 系统结构中的并行性

从计算机系统中执行程序的角度来看，并行性等级从低到高可以分为四级。它们分别是：

指令内部——一条指令内部各个微操作之间的并行。

指令之间——多条指令的并行执行。

任务或进程之间——多个任务或程序段的并行执行。

作业或程序之间——多个作业或多道程序的并行。



1.6 系统结构中的并行性

从计算机系统中处理数据的并行性来看，并行性等级从低到高可以分为：

位串字串——同时只对一个字的一位进行处理，这通常是指传统的串行单处理机，没有并行性。

位并字串——同时对一个字的全部位进行处理，这通常是指传统的并行单处理机，开始出现并行性。

位片串字并——同时对许多字的同一位(称位片)进行处理，开始进入并行处理领域。

全并行——同时对许多字的全部或部分位组进行处理。



1.6 系统结构中的并行性

并行性是贯穿于计算机信息加工的各个步骤和阶段的，从这个角度来看，并行性等级又可分为：

存储器操作并行——可以采用单体多字、多体单字或多体多字方式在一个存储周期内访问多个字，进而采用按内容访问方式在一个存储周期内用位片串字并或全并行方式实现对存储器中大量字的高速并行比较、检索、更新、变换等操作。典型的例子就是并行存储器系统和以相联存储器为核心构成的相联处理机。



1.6 系统结构中的并行性

处理器操作步骤并行——处理器操作步骤可以指一条指令的取指、分析、执行等操作步骤，也可指如浮点加法的求阶差、对阶、尾加、舍入、规格化等具体操作的执行步骤。处理器操作步骤并行是将操作步骤或具体操作的执行步骤在时间上重叠流水地进行。典型的例子就是流水线处理机。

处理器操作并行——为支持向量、数组运算，可以通过重复设置大量处理单元，让它们在同一控制器的控制下，按照同一条指令的要求对多个数据组同时操作。典型的例子就是阵列处理机。



1.6 系统结构中的并行性

指令、任务、作业并行——这是较高级的并行，虽然它也可包含如操作、操作步骤等较低等级的并行，但原则上与操作级并行是不同的。指令级以上的并行是多个处理机同时对多条指令及有关的多数据组进行处理，而操作级并行是对同一条指令及其有关的多数据组进行处理。因此，前者构成的是多指令流多数据流计算机，后者构成的则是单指令流多数据流计算机。典型的例子是多处理机。



1.6 系统结构中的并行性

2. 并行性开发的途径

- 时间重叠
- 资源重复
- 资源共享



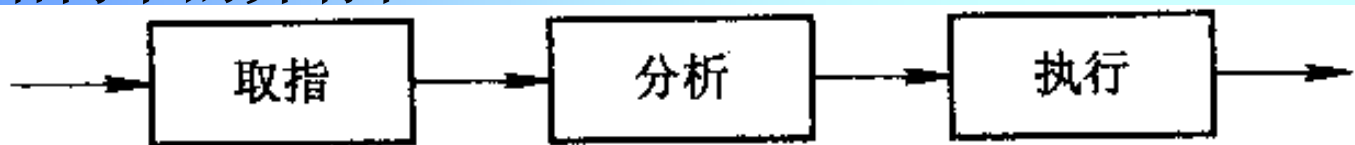
1.6 系统结构中的并行性

时间重叠

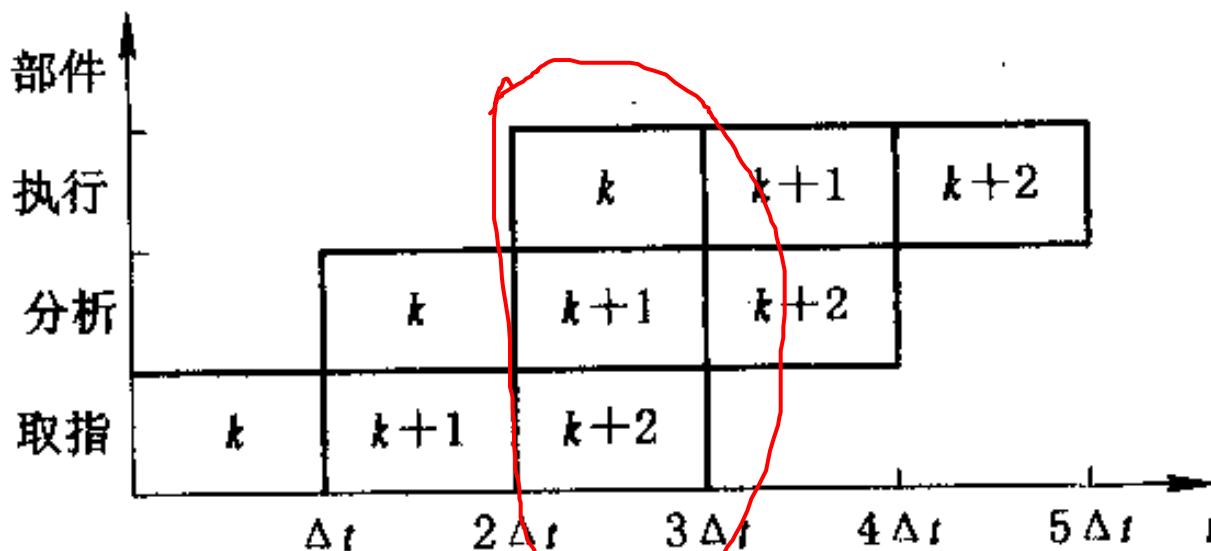
在并行性概念中引入时间因素，让多个处理过程在时间上相互错开，轮流重叠地使用同一套硬件设备的各个部分，以加快硬件周期而赢得速度。



1.6 系统结构中的并行性



(a) 指令流水线



(b) 指令在流水线各部件中流过的时间关系

图 1.7 时间重叠的例子



1.6 系统结构中的并行性

资源重复

在并行性概念中引入空间因素，通过重复设置硬件资源来提高可靠性或性能。

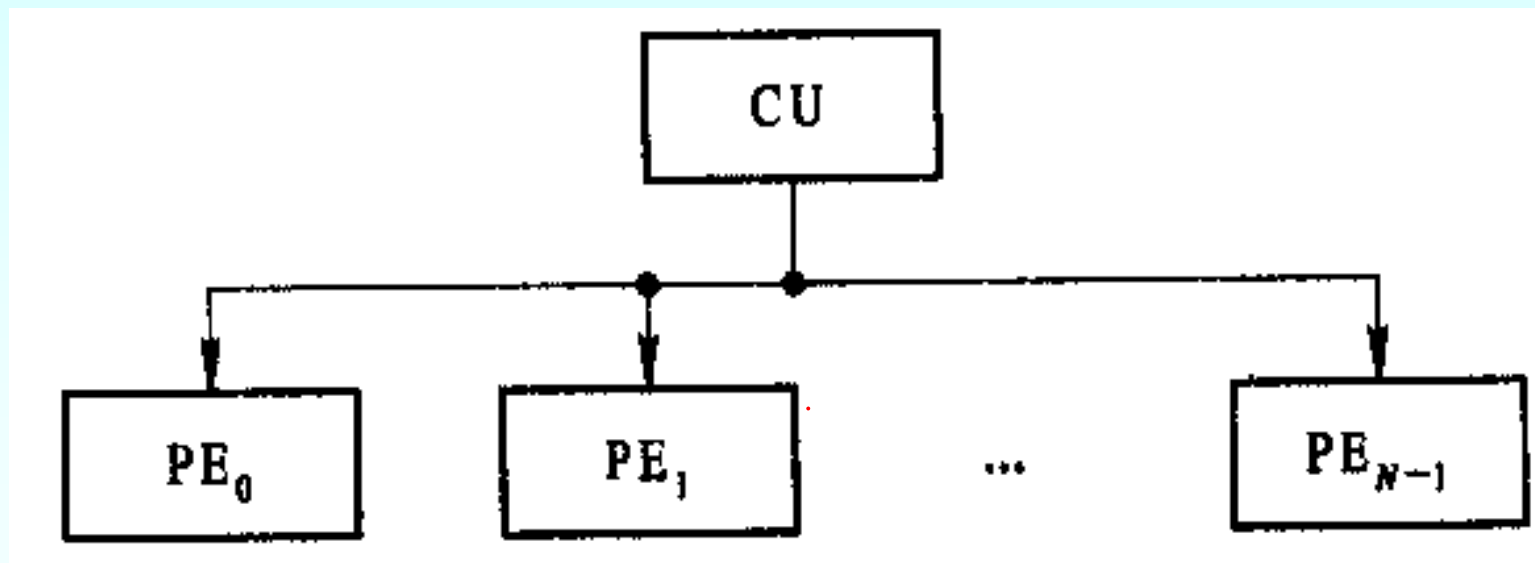


图 1.8 资源重复的例子



1.6 系统结构中的并行性

资源共享

利用软件的方法让多个用户按一定时间顺序轮流地使用同一套资源，以提高利用率，这样也可以提高整个系统的性能。

例如：多道程序分时系统。

再例如：共享主存、外设、通信线路的多处理机，计算机网络，以及分布处理系统。

资源共享不只限于硬件资源的共享，也包括软件、信息资源的共享。



1.6 系统结构中的并行性

1.6.2 并行处理系统的结构与多机系统的耦合度

1. 并行处理计算机的结构

并行处理计算机是强调并行处理的系统，除了分布处理系统外，按其基本结构特征，可以分成流水线计算机、阵列处理机、多处理机系统和数据流计算机等 4 种不同的结构。

流水线计算机主要通过时间重叠，让多个部件在时间上交错重叠地并行执行运算和处理，以实现时间上的并行。



1.6 系统结构中的并行性

阵列处理机主要通过资源重复，设置大量算术逻辑单元，在同一控制部件作用下同时运算和处理，以实现空间上的并行。相联处理机也可归属这一类。由于各个处理器(机)是同类型的且完成同样的功能，所以主要是一种对称、同构型多处理器(机)系统。阵列处理机上主要要解决好处理单元间的灵活而有规律的互连模式及互连网络的设计、存储器组织、数据在存储器中的分布，以及研制对具体题目的高效并行算法等问题。



1.6 系统结构中的并行性

多处理机系统主要通过资源共享，让共享输入/输出子系统、数据库资源及共享或不共享主存的一组处理机在统一的操作系统全盘控制下，实现软件和硬件各级上相互作用，达到时间和空间上的异步并行。它可以改善系统的吞吐量、可靠性、灵活性和可用性。多处理机系统根据各处理机是否共享主存，可分为紧耦合和松耦合两种不同类别。多处理机系统主要解决的问题是，处理机机间的互连、存储器组织等硬件结构，存储管理、资源分配、任务分解、系统死锁的防止、进程间的通信和同步、多处理机的调度、系统保护等操作系统，高效并行算法和并行语言的设计等问题。



1.6 系统结构中的并行性

2. 多机系统的耦合度

多机系统指的是多处理机系统和多计算机系统。多处理机系统与多计算机系统是有差别的。多处理机系统是由多台处理机组成的单一计算机系统，各处理机都可有自己的控制部件，可带自己的局部存储器，能执行各自的程序，它们都受逻辑上统一的操作系统控制，处理机间以文件、单一数据或向量、数组等形式交互作用，全面实现作业、任务、指令、数据各级的并行。多计算机系统则是由多台独立的计算机组成的系统，各计算机分别在逻辑上独立的操作系统控制下运行，机间可以互不通信，即使通信也只是经通道或通信线路以文件或数据集形式进行，实现多个作业间的并行。



1.6 系统结构中的并行性

为了反映多机系统中各机器之间物理连接的紧密程度和交叉作用能力的强弱，引入耦合度概念。多机系统的耦合度，可以分为**最低耦合、松散耦合和紧密耦合**等。

各种脱机处理系统是最低耦合系统 (Least Coupled System)，其耦合度最低，除通过某种中间存储介质之外，各计算机之间并无物理连接，也无共享的联机硬件资源。例如，独立外围计算机系统由主机和外围计算机组成，后者脱机工作，只通过磁带、软盘或纸带等对主机的输入/输出提供支持。



1.6 系统结构中的并行性

如果多台计算机通过通道或通信线路实现互连，共享某些如磁带、磁盘等外设，则称为松散耦合系统。

如果多台计算机之间通过总线或高速开关互连，共享主存，则称为紧密耦合系统。



1.7 计算机系统的分类

1. 弗林（Flynn）分类法

1966年由 **Michael.J.Flynn** 提出。

按照指令流和数据流的多倍性特征对计算机系统进行分类。

指令流：机器执行的指令序列。

数据流：由指令流调用的数据序列，包括输入数据和中间结果。

多倍性：在系统性能瓶颈部件上同时处于同一执行阶段的指令或数据的最大可能个数。



1.7 计算机系统的分类

指令流 \ 数据流	单	多
	单	多
单	SISD	SIMD
多	MISD	MIMD



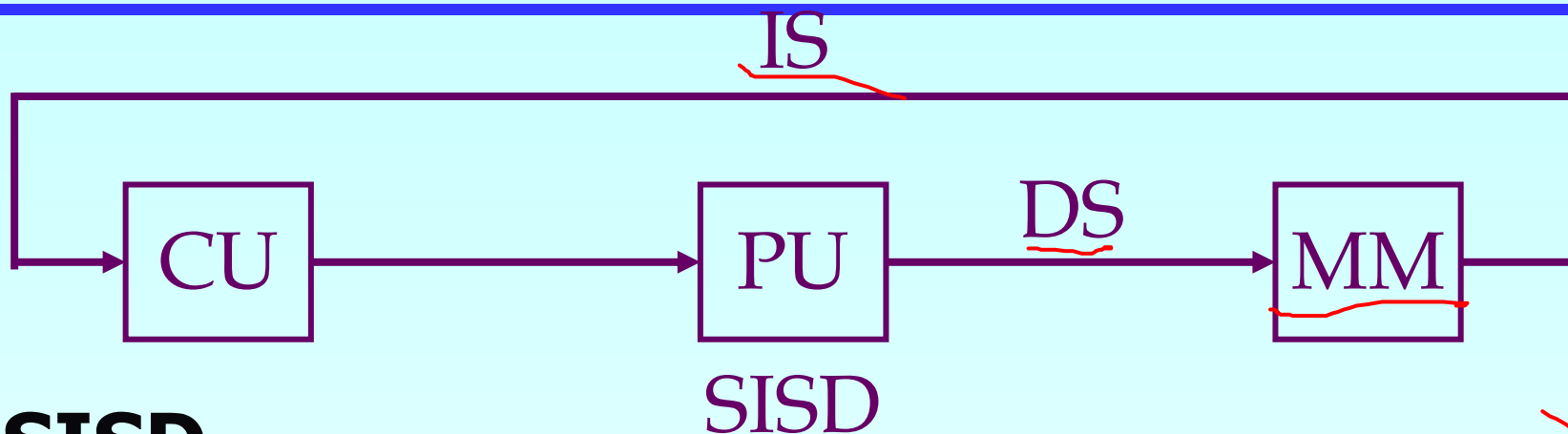
1.7 计算机系统的分类

四种类型

- 单指令流单数据流 **SISD** (Single Instruction Single Datastream);
- 单指令流多数据流 **SIMD** (Single Instruction Multiple Datastream);
- 多指令流单数据流 **MISD** (Multiple Instruction Single Datastream);
- 多指令流多数据流 **MIMD** (Multiple Instruction Multiple Datastream)



1.7 计算机系统的分类



SISD:

它每次只对一条指令译码，并只对一个操作部件分配数据。

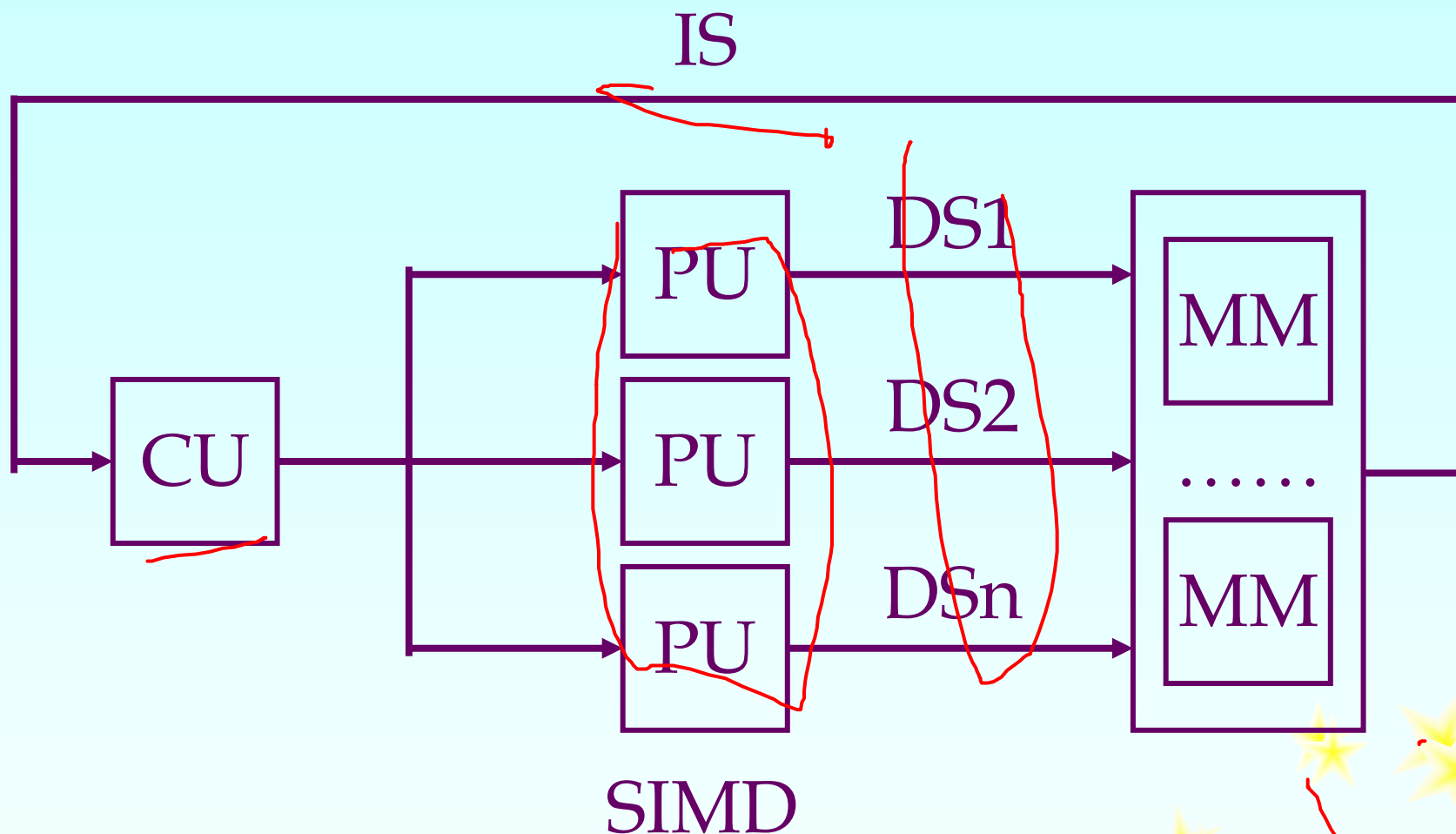
典型单处理机，包括：

单功能部件处理机：**IBM 1401, VAX-11**

多功能部件处理机：**IBM360/91, 370/168, CDC6600**



1.7 计算机系统的分类



1.7 计算机系统的分类

SIMD

多个**PU**按一定方式互连，在同一个**CU**控制下，多个各自的数据完成同一条指令规定的操作；从**CU**看，指令顺序（串行）执行，从**PU**看，数据并行执行。

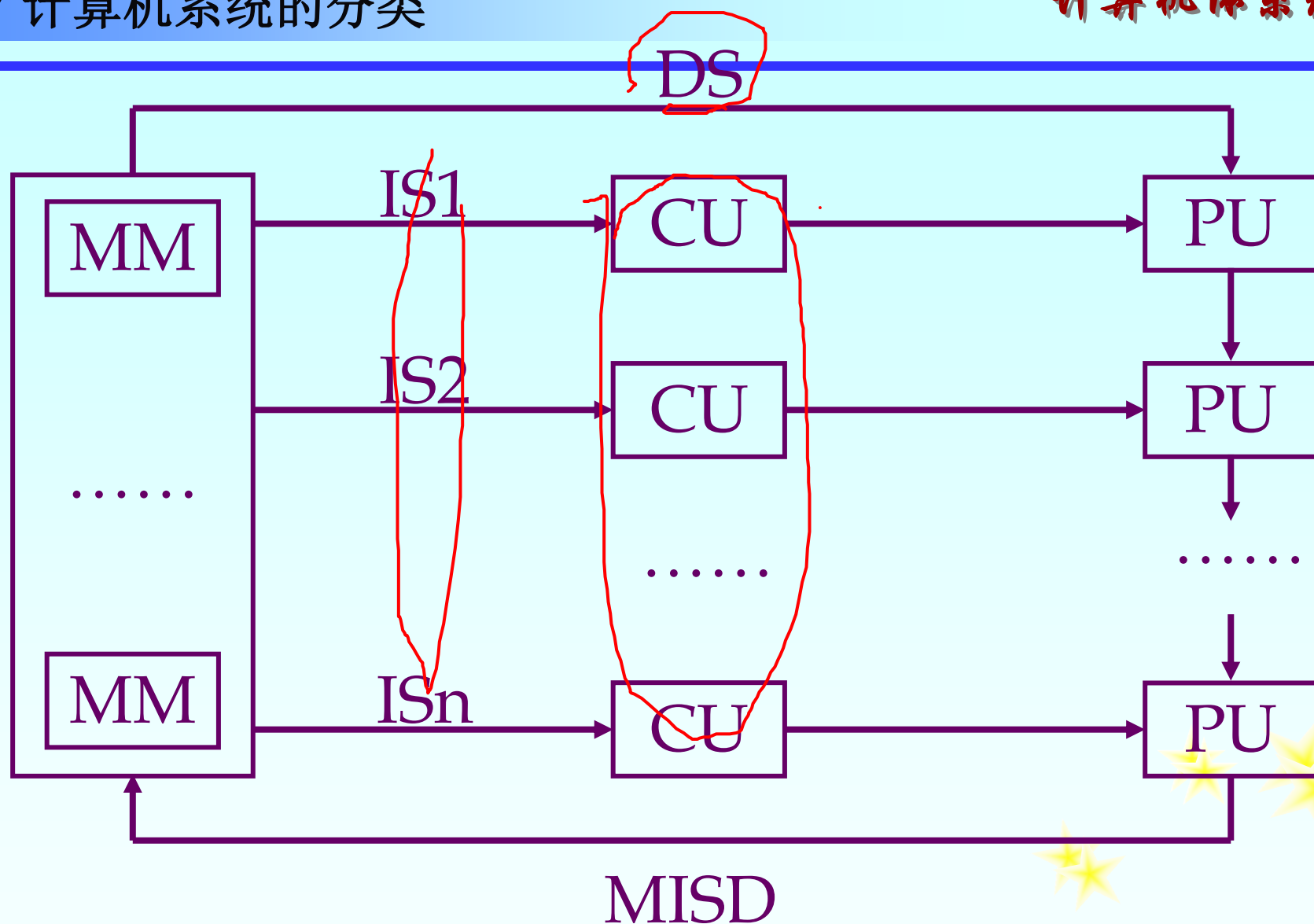
并行处理机、阵列处理机、向量处理机、相联处理机、....

数据全并行：**ILLIAC IV**、**PEPE**、**STAR100**、**TI-ASC**、**CRAY-1**。

数据字并位串：**STARAN**、**MPP**、**DAP**。



1.7 计算机系统的分类



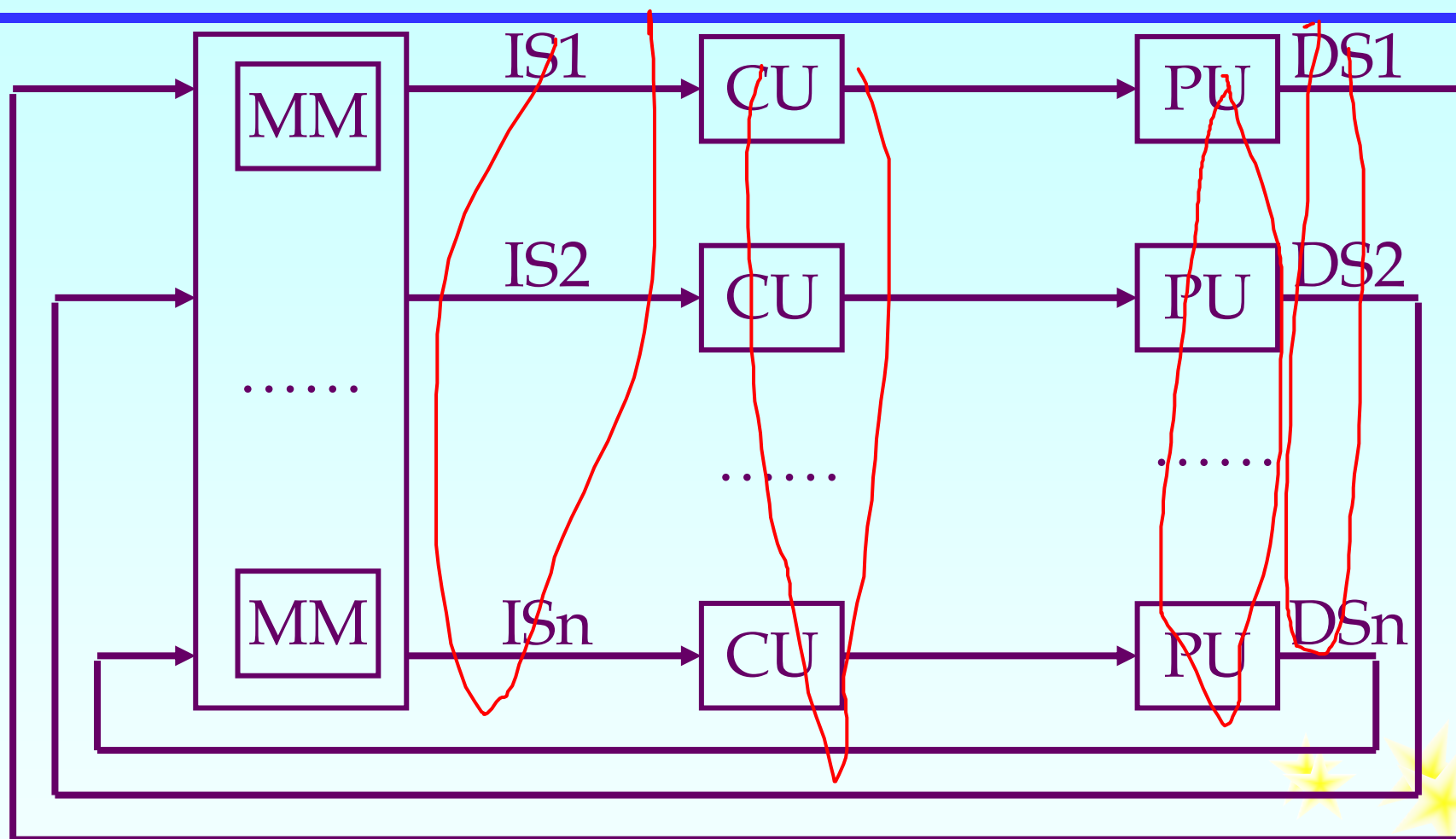
1.7 计算机系统的分类

MISD:

几条指令对同一个数据进行不同处理，因为它要求系统在指令级上并行，而在数据级上又不并行，这是不太现实的。但现在也有些学者有不同的看法，在有些文献中将超级标量机以及超长指令字计算机等看作是**MISD**类型。



1.7 计算机系统的分类



MIMD



1.7 计算机系统的分类

MIMD:

能实现作业、任务、指令、数组各级全面并行的多机系统，它包括了大多数多处理机及多计算机系统。

松散耦合：**IBM3081、IBM3084、UNIVAC-1100/80、Cm***

紧密耦合：**D-825、Cmmp、CRAY-2**



1.7 计算机系统的分类

表 按Flynn法对计算机分类举例

类 型		计算机的型号
SISD	单功能部件	IBM 701, IBM 1401, IBM 1620, IBM 7090, PDP-11, VAX-11/780
	多功能部件	IBM 360/91, IBM 370/168 UP, CDC 6600
SIMD	数据全并行	ILLIAC-N, PEPE, BSP, CDC STAR-100, TI-ASC, FPS AP-120B, FPS-164, IBM 3838, CRAY-1, CYBER 205, Fujitsu VP-200, CDC-NASF, B-5000
	数据位片串字并	STARAN, MPP, DAP
MIMD	松耦合	IBM 370/168 MP, UNIVAC 1100/80, IBM 3081/3084, Cm*
	紧耦合	Burroughs D-825, C.mmp, CRAY-2, S-1, CRAY-XMP, Denelcor HEP

1.7 计算机系统的分类

Flynn分类法得到广泛应用

SISD、**MIMD**、**SIMD**...

Flynn分类法的主要缺点：

(1) 分类太粗：例如，在**SIMD**中包括有多种处理机，对流水线处理机的划分不明确，标量流水线为**SISD**，向量流水线为**SIMD**。

(2) 根本问题是把两个不同等级的功能 并列对待；通常，数据流受指令流控制，从而造成**MISD**不存在。

(3) 非冯计算机的分类？其他新型计算机的分类？



1.7 计算机系统的分类

2. 库克分类法

1978年由D. J. Kuck提出。

按指令流和执行流分类。

四种类型：

(1) 单指令流单执行流**SISE** (Single Instruction Single Executionstream); 典型的单处理机。

(2) 单指令流多执行流**SIME** (Single Instruction Multiple Executionstream); 多功能部件处理机、相联处理机、向量处理机、流水线处理机、超流水线处理机、超标量处理机、**SIMD**并行处理机。



1.7 计算机系统的分类

(3) 多指令流单执行流MISE (Multiple Instruction Single Executionstream);
多道程序系统。

(4) 多指令流多执行流MIME (Multiple Instruction Multiple Executionstream);
典型的多处理机。



1.7 计算机系统的分类

主要缺点：

有些系统，如分布处理机等，没有总控制器。

分类级别太低，没有处理机级和机器级。

分类太粗，如**SIME**中包含了多种类型的处理机。



3. 冯泽云分类法

1972年美籍华人冯泽云提出。

用最大并行度来对计算机系统进行分类。

最大并行度：计算机系统在单位时间内能够处理的最大二进制位数。假设同时处理的字宽为**n**，位宽为**m**，则最大并行度定义为：

$$P_m = m \times n$$



1.7 计算机系统的分类

平均并行度：假设每个时钟周期 t_i 内能同时处理的二进位数为 B_i ，则 T 个时钟周期内的平均并行度定义为：

$$P_n = \frac{\sum_{i=1}^T B_i \cdot t_i}{T}$$

表示方法：处理机名 (n, m)

P_m = 等于整数 n 和 m 确定的矩形面积。



1.6 系统结构中的并行性及计算机系统的分类

四种类型

(1) 字串位串 **WSBS****(Word Serial and Bit Serial)**每次只处理一个字中的一位。串行计算机; $m=1, n=1$;例如: **EDVAC(1, 1)**(2) 字串位并 **WSBP****(Word Serial and Bit Parallel)**每次只处理一个字中的n位。传统的单处理机; $m=1, n>1$;例如: **Pentium(32, 1)**

1.7 计算机系统的分类

(3) 字并位串 WPBS**(Word Parallel and Bit Serial)**一次处理 m 个字中的 1 位。（位片处理）并行计算机、**MPP**、相联计算机； **$m > 1, n = 1;$** 例如：**MPP(1, 16384), STARAN(1, 256), DAP(1, 4096)**

(4) 字并位并WPBP (Word Parallel and Bit Parallel)

一次处理m个字，每个字为n位。

全并行计算机； $m > 1, n > 1$;

例如：ILLIAC IV(64,64),

ASC(64,32), PEPE(32,288),

Cmmp(16,16)

主要缺点：

仅考虑了数据的并行性，没有考虑指令、任务、作业的并行。



1.7 计算机系统的分类

4. 汉德勒分类法

由**Wolfgan Hindler**于**1977**年提出，又称为**ESC (Erlange Classification Scheme)**分类法。

根据并行度和流水线分类，计算机的硬件结构分成三个层次（程序级、操作级、逻辑级），并分别考虑它们的可并行性和流水处理程度。



1.7 计算机系统的分类

为了表示流水线，采用：

$$t(\text{系统型号}) = (k \times k', d \times d', w \times w')$$

其中：

k' 表示宏流水线中程序控制部件的个数

d' 表示指令流水线中算术逻辑部件的个数

w' 表示操作流水线中基本逻辑线路的套数



1.7 计算机系统的分类

例如：Cray1有1个CPU，12个相当于ALU或PE的处理部件，最多8级流水线，字长为64位，可以实现1~14位流水线。表示为：

$$t(\text{Cray1}) = (1, 12 \times 8, 64 \times (1 \sim 14))$$

又例如：

$$t(\text{PEPE}) = (1 \times 3, 288, 32)$$

$$t(\text{TI ASC}) = (1, 4, 64 \times 8)$$



1.7 计算机系统的分类

5. 其他分类法

如从对执行程序或指令的控制方式上分类

控制驱动的控制流方式

数据驱动的数据流方式

需求驱动的规约方式

模式驱动的匹配方式



本章重点

计算机系统的层次结构

计算机系统结构的定义及研究对象

计算机系统结构、组成、实现三者的区别与相互联系。

计算机系统设计的主要方法（由中间开始设计）

Amdahl定律

CPU性能公式



解决软件可移植性的三种方法

透明性、系列机、兼容性、模拟与仿真
等基本概念

系统结构中的并行性

并行性开发的途径

并行处理系统的结构和多机系统的耦合
度

计算机系统的分类方法（弗林分类法）



术语解释

翻译、解释、计算机系统结构、透明性、软件兼容、模拟、仿真、并行性、系列机、兼容机、时间重叠、紧耦合系统



作业

1-1

1-6

1-8

1-9

1-10

1-11

