



# 加速器控制技术

王春红

高能所加速器中心自控组

# 引言

---

- ◆ 加速器控制是通过计算机和接口装置对组成加速器的元器件如磁铁电源、真空、高频等实施控制和监测。
- ◆ 控制是加速器的神经网络,确保加速器设备间协调运作。
- ◆ 控制涵盖的内容广: 计算机、网络、操作系统、数据库、应用软件、设备接口及驱动、信号处理和反馈、电子学等多方面的技术
- ◆ 加速器控制属于大型的控制系统,集成度要求高

# 主要内容

---

- ◆ 第一章 控制系统的基本概念
- ◆ 第二章 加速器控制技术
  - 2.1 总线及I/O接口
  - 2.2 电磁兼容与接地
  - 2.3 网络与数据通讯
  - 2.4 时间信息的获取与传输
  - 2.5 实时多任务系统
  - 2.6 控制软件设计
  - 2.7 系统可靠性
- ◆ 第三章 分布式控制系统实例--EPICS系统
- ◆ 第四章 CSNS控制系统设计

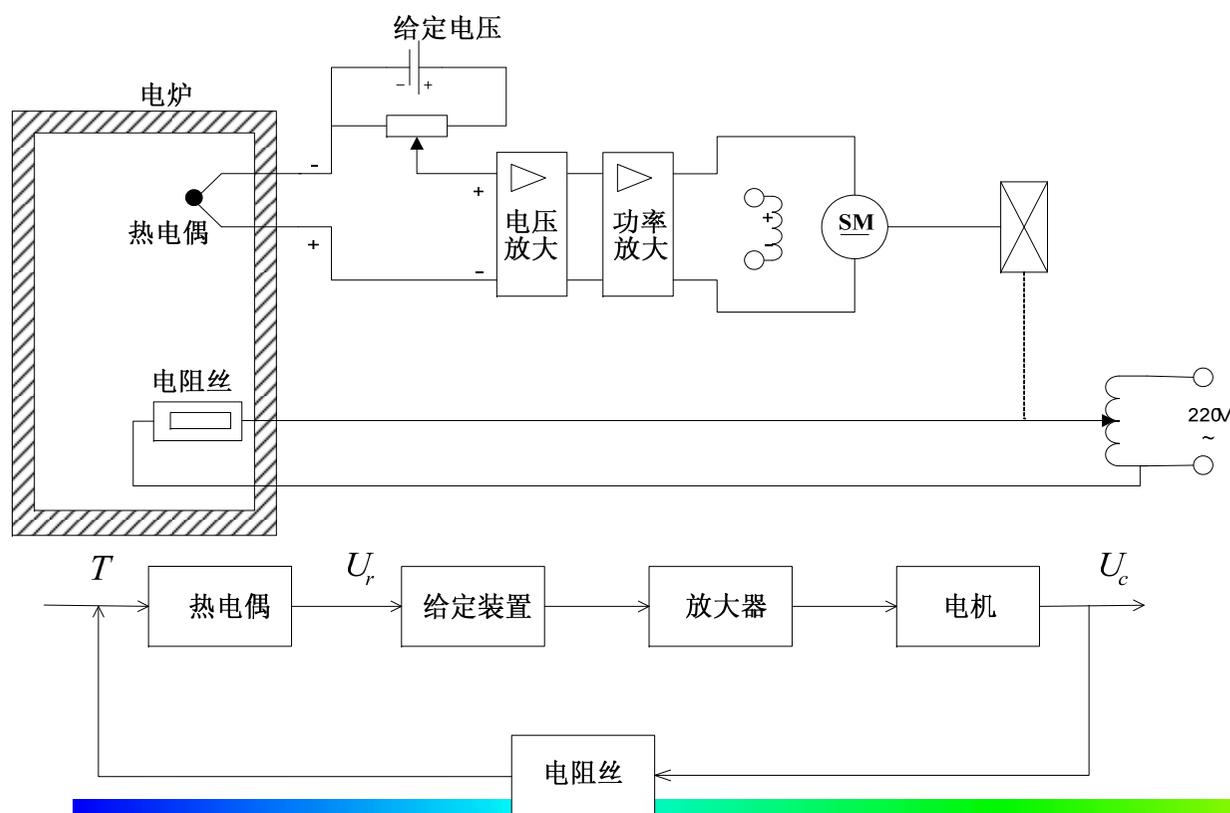
# 第一章 控制系统的基本概念

---

1. 模拟自动控制与计算机控制的区别
  - ① 模拟自动控制
  - ② 计算机控制
2. 现代控制系统的组成
3. 控制系统的体系结构
  - ① 嵌入式系统
  - ② 机箱式系统
  - ③ 分布式系统
4. 控制系统的工作方式
  - ① 开环控制
  - ② 闭环控制
5. 分布式控制系统

# 1.1 模拟自动控制

- ◆ 所谓模拟自动控制，是指在没有人直接参与的情况下，利用外加的设备或装置（称控制装置或控制器），使机器、设备或生产过程（统称被控对象）的某个工作状态或参数（即被控量）自动地按照预定的规律运行。



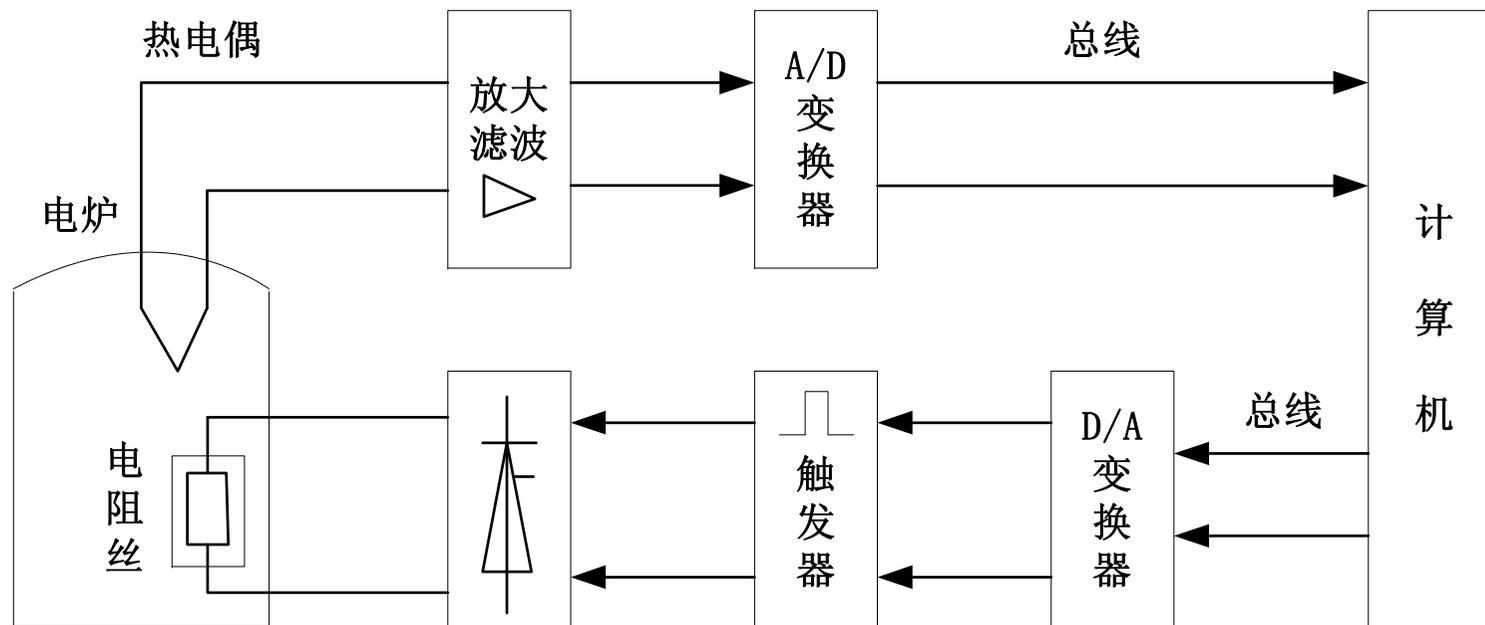
## ❖ 分析电炉温度控制系统原理

❖ 系统的被控对象为电炉，被控量为电炉温度，输入量为电阻丝的电压。

❖ 炉温实际值由热电偶检测，并转换成电压，经放大后给电动机提供一定的工作电压，驱动电动机通过改变电阻丝的工作电压，从而改变电阻丝中的电流大小，达到控制炉温的目的。

# 1.2 计算机控制

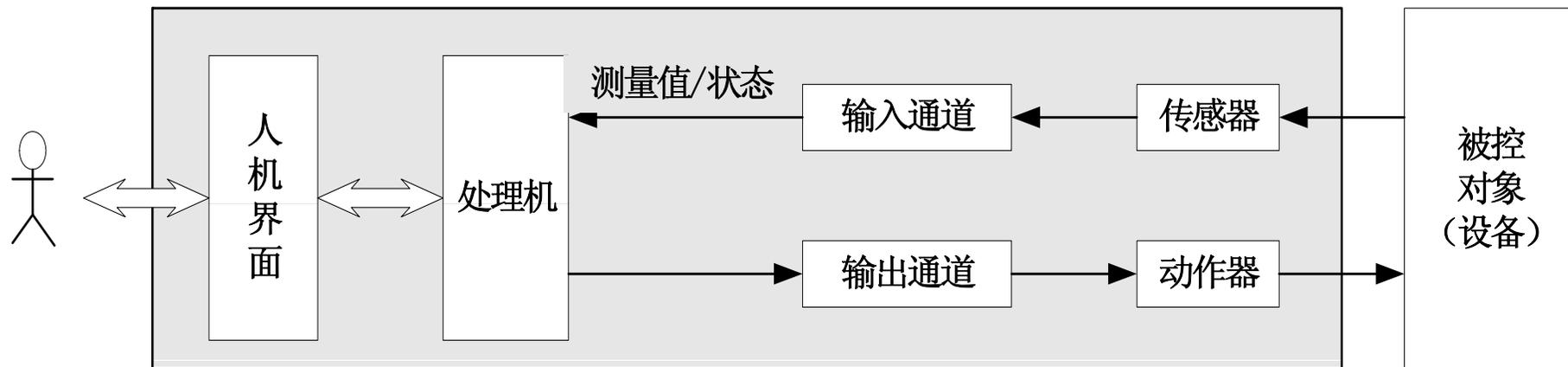
- ◆ 随着微处理器的出现和微电子学、计算机及网络技术的发展，现代控制系统发展成为完全数字化的、以计算机为核心的系统，通称为计算机控制。
- ◆ 分析电阻炉微型计算机温度控制系统原理：电阻丝通过晶闸管主电路加热，炉温期望值用计算机键盘预先设置，炉温实际值由热电偶检测，并转换成电压，经放大、滤波后，由A/D变换器将模拟量变换为数字量送入计算机，在计算机中与所设置的温度期望值比较后产生偏差信号，计算机便根据预定的控制算法计算出相应的控制量，再经D/A变换器变换成电流，通过触发器控制晶闸管导通角，从而改变电阻丝中电流大小，达到控制炉温的目的。
- ◆ 该系统既有精确的温度控制功能，还有实时屏幕显示和打印功能，以及超温、极值和电阻丝、热电偶损坏报警等功能。



电阻炉微型计算机温度控制系统原理示意图

## 2. 现代计算机控制系统的基本组成

- ◆ 输入/输出模块，提供输入/输出通道 (I/O Channels)。输入通道通过传感器与被控设备接口取得设备的状态信息，如温度、压力、流量、磁场强度以及各种开关状态。输出通道通过动作器与被控设备接口向设备传送控制指令或设定各种控制量。
- ◆ 以微处理器(Microprocessor)为核心的处理机。
- ◆ 人机界面(Human-Machine Interface,HMI).操作者通过人机界面对被控对象进行操作与调整，了解被控对象的状态。



# 3. 控制系统体系结构

◆ 根据规模的不同，控制系统大致可以分为：

- ① 嵌入式系统
- ② 机箱式系统
- ③ 分布式系统

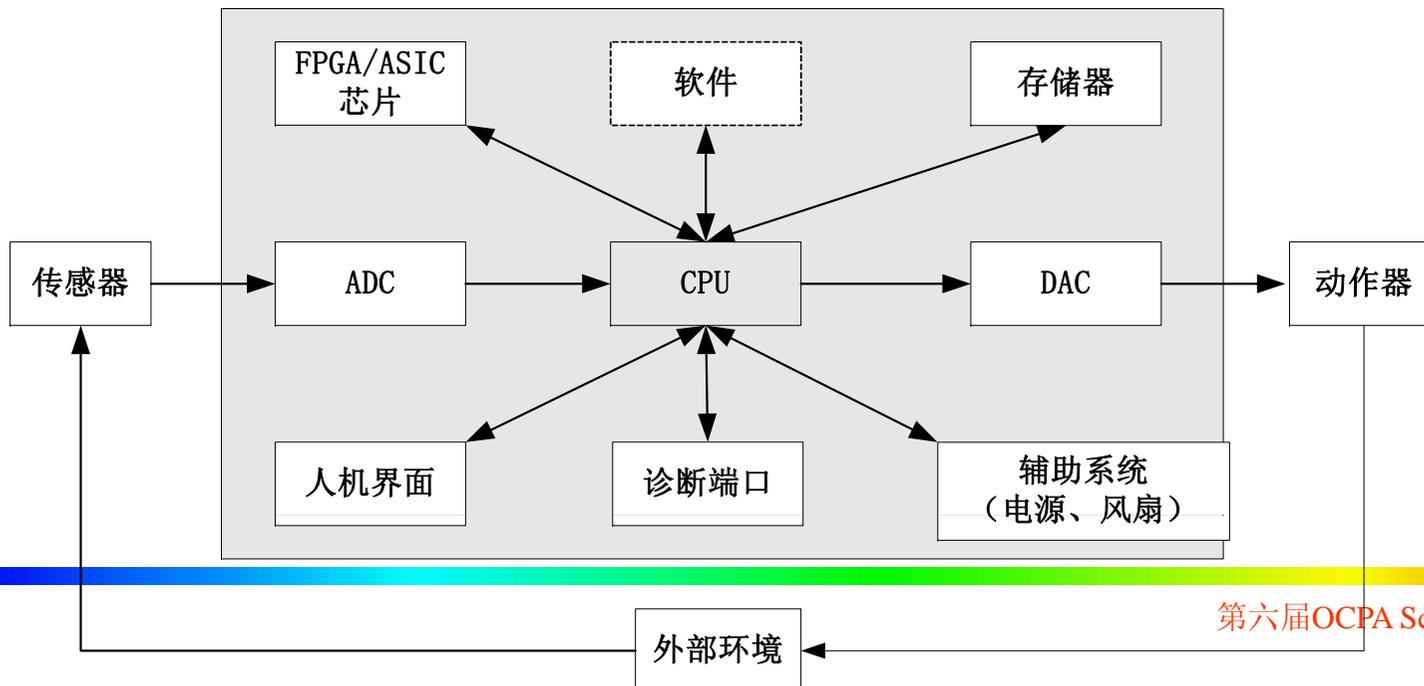
◆ 嵌入式系统本身是一个相对模糊的定义。目前国内一个普遍被认同的定义是：以应用为中心、以计算机技术为基础、软件硬件可裁剪、适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。

◆ 虽然嵌入式系统是近几年才风靡起来的，但是这个概念并非新近才出现。从20世纪七十年代单片机的出现到今天各式各样的嵌入式微处理器，微控制器的大规模应用，嵌入式系统已经有了近30年的发展历史。

◆ 所谓嵌入式系统 (**Embedded System**)是一种“嵌”在被控设备中的控制系统

# 3.1 嵌入式系统

- ◆ 一般而言，嵌入式系统的构架可以分成四个部分：处理器、存储器、输入输出（I/O）和软件。
- ◆ 从硬件结构来看，嵌入式系统主体通常包括在一个电路模块中或在一个芯片中（**System-on-Chip, SoC**）。
- ◆ 从软件结构来看，低端的嵌入式系统通常采用“超循环”方法处理多任务，较复杂的系统则在实时操作系统（内核）管理下运行。
- ◆ 人机界面：多数是比较简单的小型显示屏/指示灯和按键/旋钮
- ◆ 嵌入式系统的硬件结构是根据具体应用而设计的，不便于改动，适合于批量生产的产品。例如：手机、打印机、复印机、洗衣机、微波炉、电视和**DVD**等各种智能化电器。



## 3.2 机箱式系统

- ◆ 系统由若干模块（1个或多个处理机模块，图形处理模块以及I/O模块等）组成，它们通过机箱背板总线（例如VME和CompactPCI）构成紧耦合的系统。这种系统在实时操作系统管理下运行，实时性和可靠性都很高。
- ◆ 低端的机箱式系统，通常采用台式PC机或者PC工控机，运行Windows。
  - ❖ 优点：价格低廉，有很好的窗口图形界面和丰富的商业软件支持
  - ❖ 缺点：系统实时性和可靠性较低
- ◆ 高端的机箱式系统采用背板总线，实行并行的、所有模块共享数据通道的数据传输方式。
  - ❖ 优点：机械/电气性能好，可靠性和实时性高
  - ❖ 缺点：造价高，背板总线数据传输速率有限制
- ◆ 高速串行总线和交换式结构的机箱式系统（新发展）
  - ❖ 机箱中设有专门供交换控制器使用的插槽，通过交换控制器来实现各个模块之间的点到点的串行数据传输。
  - ❖ 利用网络交换技术，极大地提高了模块之间的数据传输速率
  - ❖ VITA46/VPX、ATCA、microTCA

# 3.3 分布式系统

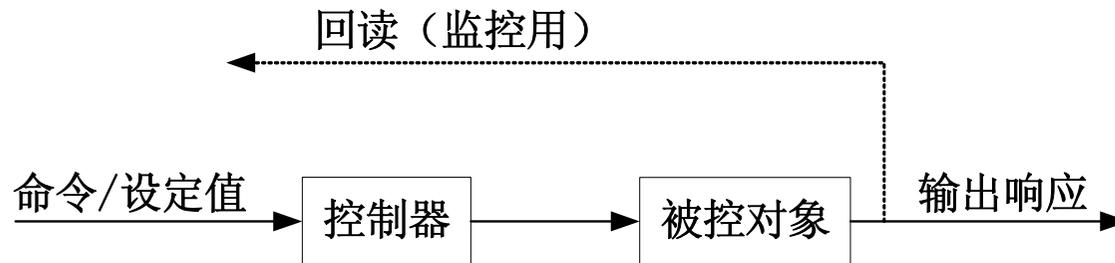
- ◆ 分布式控制系统 ( **Distributed Control System,DCS**)是基于网络的中、大规模控制系统。
- ◆ 在分布式控制系统中，系统分为3层：
  - ❖ 管理层、前端控制层以及设备控制层
  - ❖ 各层次计算机通过网络通信交换数据和控制信息
- ◆ 不同规模的控制系统的特点

	嵌入式系统	机箱系统	分布式系统
控制范围	单个设备	实验室范围	10~1000米控制半径
I/O通道数	10~100量级	10~1000量级	100~100 000量级
硬件系统	系统在1块电路模块上	紧耦合/背板总线系统	松耦合/网络系统
系统软件	低端应用：无操作系统 高端应用：实时操作系统	实时操作系统	OPI：桌面操作系统 IOC：实时操作系统

# 4 开环控制和闭环控制

## 4.1 开环控制 (open-loop system)

开环控制是一种简单直接的控制方法即将命令或设定值输入到控制器，直接在被控对象产生所希望的输出响应



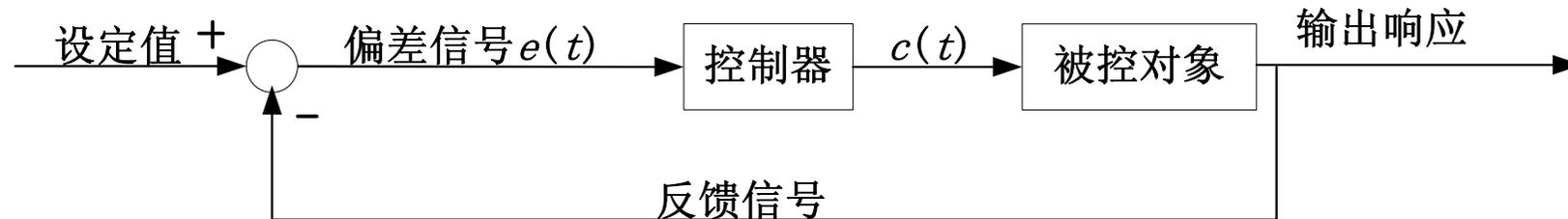
开环控制的特点：逻辑控制+设定值控制

逻辑控制：

- 顺序控制：使各个被控对象按照预定的顺序运行，如电源的启动/关机过程
- 条件控制：使各个被控对象在设定的条件下运行，如联锁系统
- 定时控制：使各个被控对象在设定的定时关系下运行

## 4.2 闭环控制（反馈控制/自动控制）

- ◆ 在闭环控制系统（**closed-loop system**）中，负反馈环路将被控对象的输出响应作为偏差信号，送到控制器的输入端与设定值比较（相减），将比较结果送到控制器作进一步的处理，控制器通过某种算法来调节系统的输出响应。



### ◆ 典型的闭环控制比较

- ❖ 比例（**Proportional**）控制：

$$c(t) = Pe(t)$$

- ❖ 比例积分（**PI**）控制：

$$c(t) = Pe(t) + I \int e(t) dt$$

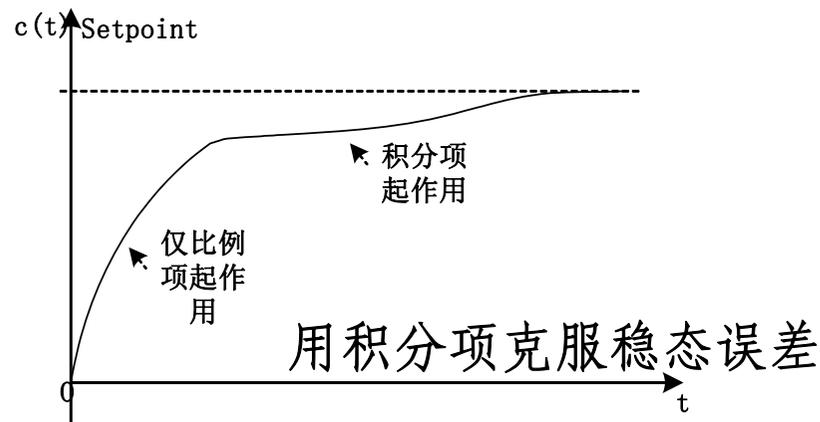
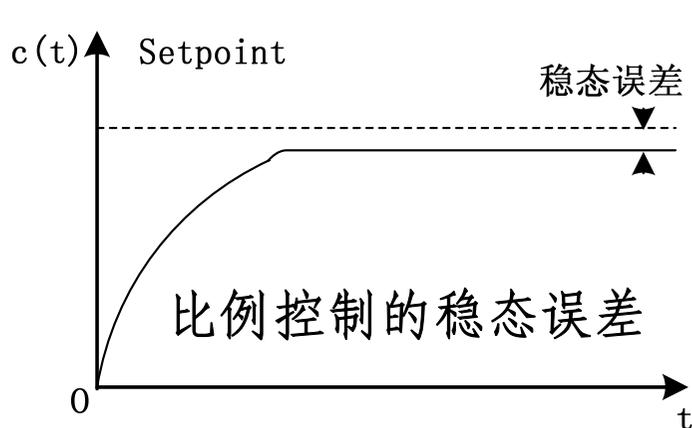
- ❖ 比例微分（**PD**）控制：

$$c(t) = Pe(t) + D \frac{de(t)}{dt}$$

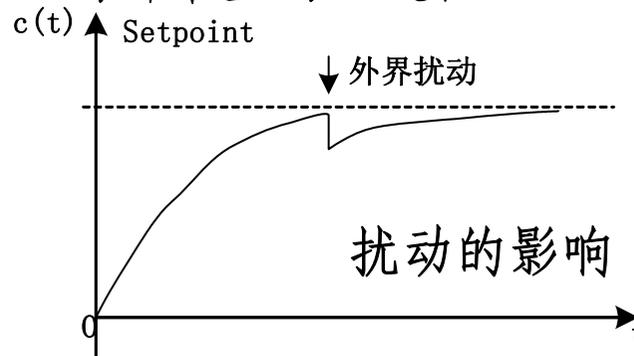
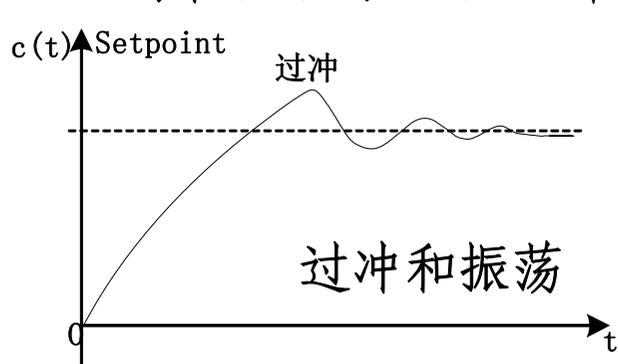
- ❖ 比例-积分-微分控制算法：

$$c(t) = Pe(t) + I \int e(t) dt + D \frac{de(t)}{dt}$$

## 4.2 典型的闭环控制作用比较

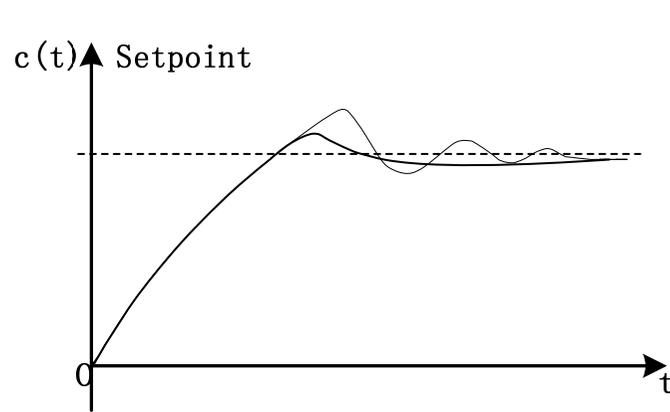


- ◆ 比例控制的静态特性不好，有稳态误差。积分是一种累积效应，只要有误差存在，积分项就不断起作用，直到达到所希望的设定值。

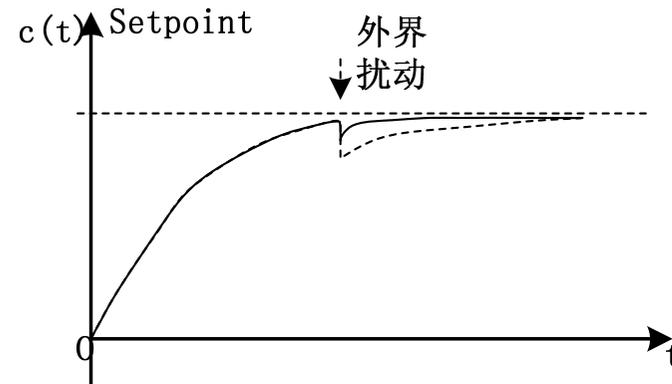


- ◆ 比例控制的动态特性也不好，会发生过冲和振荡。微分项反映输入偏差变化的速率，当突然发生偏差变化速率大的扰动时，当偏差刚出现，就可以根据偏差变化的速率，乘以系数D，提前给出大的调节量，抵消掉偏差。

## 4.2 典型的闭环控制比较



对过冲的抑制



对扰动的抑制

- ◆ 比例微分 (PD)控制:加入微分非线性项
- ◆ 比例-积分-微分控制算法: 对于特定的应用, 如何选择P、I和D的系数
- ◆ 研究闭环控制的算法称为自动控制理论, 包括:
  - ❖ 系统建模即系统的数学模型。
  - ❖ 信号滤波, 用来提取系统的状态变量和滤除随机噪声的干扰
  - ❖ 系统稳定性分析

# 5. 分布式控制系统

## ◆ 分布式控制系统:

- ❖ 使用网络，使得整个系统在地理位置上分布开来，多台计算机共同完成控制任务
- ❖ 各个节点分配不同的功能，同一层次的计算机在功能上具有互换性
- ❖ 设备的控制和监测的实时作业分布在前端计算机上（**Front-end computer**）执行

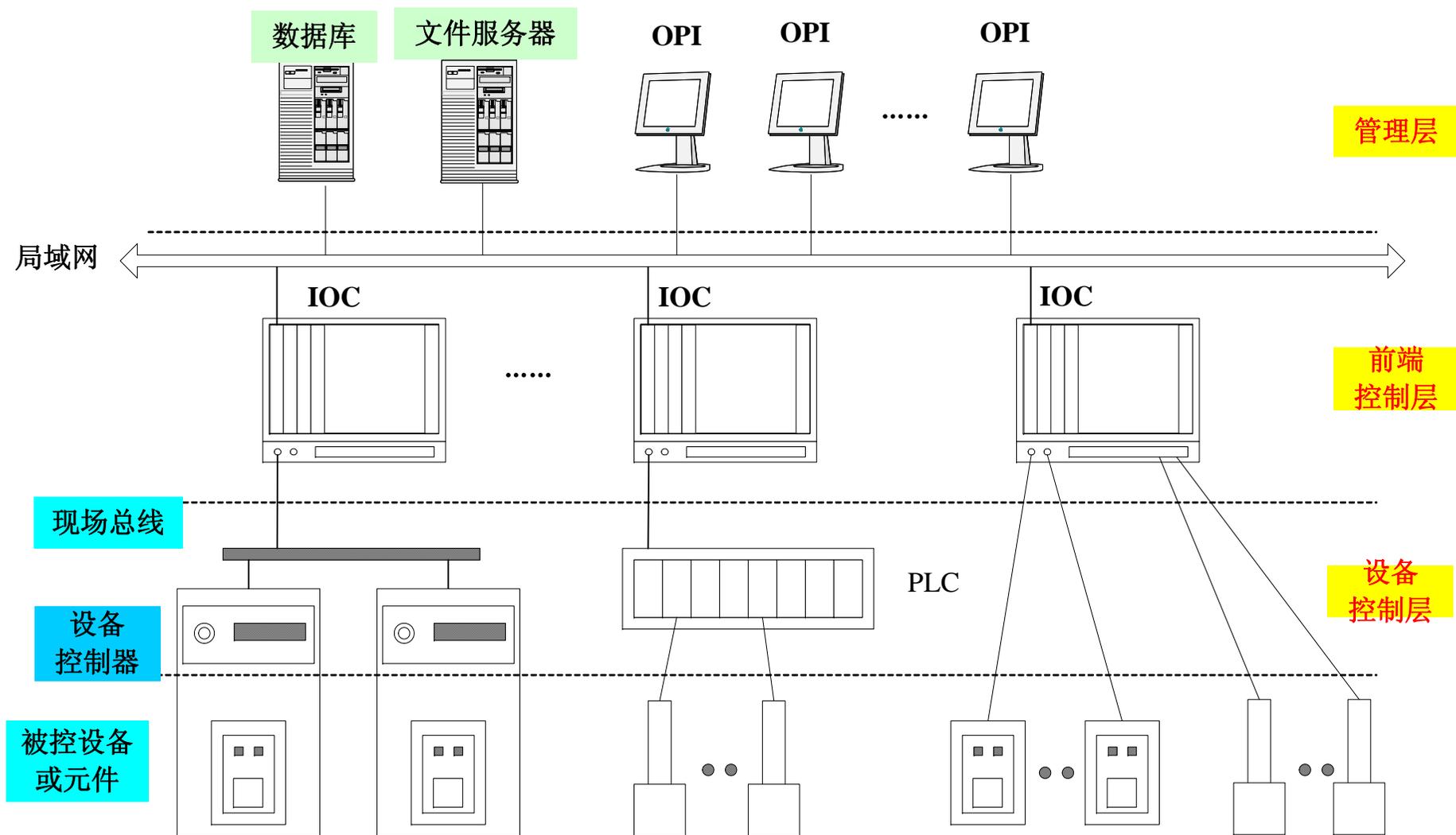
## ◆ 分布式系统是在网络协议（**TCP/IP**）的基础上，靠更高层（**OSI模型的第5~7层**）协议建立的，有以下2个特点

- ❖ 分离性：系统内有大量的独立自洽的可以相互通信的处理机
- ❖ **透明性**：对使用者而言，整个系统如同一个单一处理机系统一样透明

# 5. 分布式控制系统

- ◆ 目前国内外加速器控制系统普遍采用分布式控制系统的标准模型：两层/三层体系结构
- ◆ 管理层位于系统最上层。它通过操作员接口计算机 (Operator Interface, OPI) 来监控系统状态、发送命令或设置值和提供报警信息。管理层还包括数据库和文件服务器，数据库用来收集和及管理整个控制系统的在线数据；文件服务器用于存放控制系统软件。管理层的设备一般不要求实时性，满足要求即可
- ◆ 前端控制层 (Front-end Controls) 通过网络接受管理层的控制。前端控制计算机又称为输入输出控制机 (Input/Output Controller, IOC)。
  - ❖ IOC 通常运行实时操作系统承上启下，它保存动态数据、负责过程控制并实现控制算法，是控制任务的主要执行者。
- ◆ 设备控制层：根据被控设备的种类，结构灵活多样：
  - ❖ 可采用嵌入式设备控制器通过现场总线或串行通信与设备相连
  - ❖ 采用可编程控制器 PLC, 对子系统的过程控制和监测
  - ❖ 通过位于 IOC 内的 I/O 模块直接控制一些设备和元件

# 分布式控制系统基本结构



# 第二章 加速器控制技术

---

**2.1** 总线及I/O接口

**2.2** 电磁兼容与接地

**2.3** 网络与现场总线

**2.4** 时间信息的获取与传输

**2.5** 实时多任务系统

**2.6** 控制软件设计

**2.7** 系统可靠性

## 第二章：加速器控制技术

- ◆ 国内外加速器控制系统普遍采用分布式控制系统的标准模型：两层/三层体系结构
- ◆ 前端控制计算机是控制系统的核心，它由处理机模块、存储模块和I/O模块组成。
- ◆ 控制计算机普遍采用总线结构，背板总线是现代计算机系统结构的核心。
- ◆ 本节**2.1**重点介绍与加速器控制相关的技术
  - ❖ 介绍背板总线的基本知识，**VME总线**和**CompactPCI总线**比较
  - ❖ I/O接口

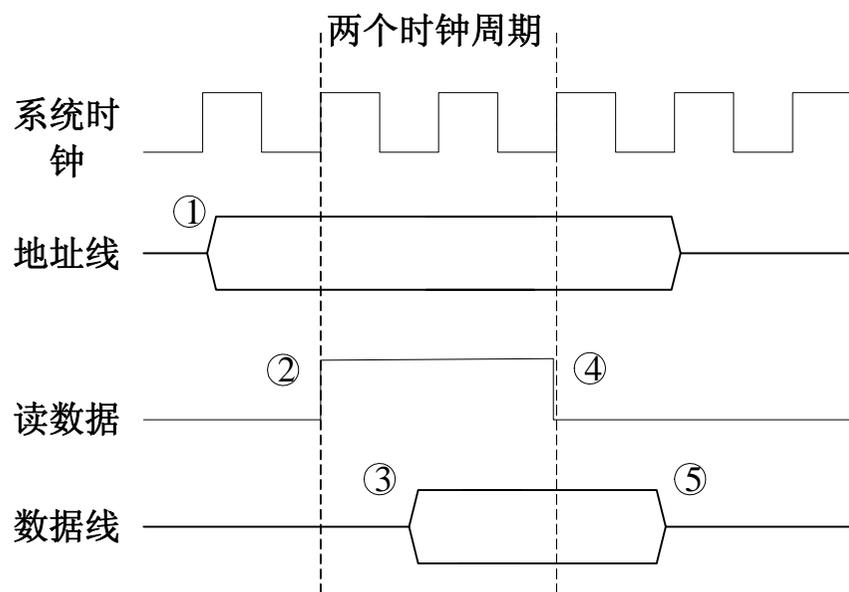
## 2.1 总线及I/O接口

- ◆ 总线的基本知识:
- ◆ 背板总线是由印刷电路板和插槽组成的并行总线，安装在机箱背面。优点：
  - ◆ 具有可靠的机械结构且便于正面插拔
  - ◆ 便于模块的设计和系统的扩充
  - ◆ 垂直的电路板使得系统易于强制通风冷却，提高了系统的可靠性
- ◆ 从概念上来说，总线由三类信号线组成：
  - ❖ 地址线：用来选择要存取数据的存储器的地址或I/O寄存器地址
  - ❖ 数据线
  - ❖ 各种控制线：用来控制数据传输，（从模块）发送中断请求，（主模块）发送要求获得总线控制权的仲裁请求以及对各种请求的响应等
  - ❖ 各种电源（+5V，+/-12V）线和地线
  - ❖ 用户自定义或将来扩充

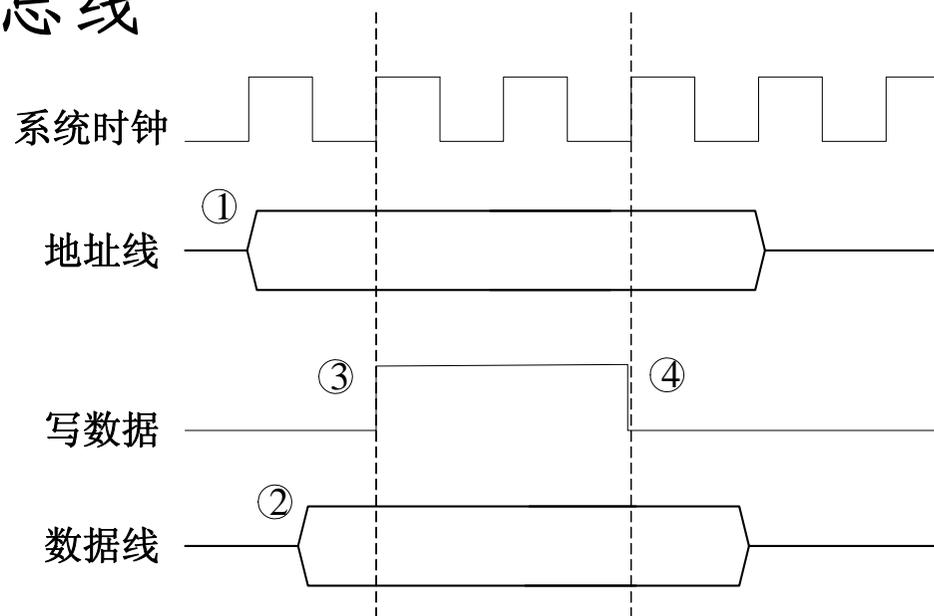
# 2.1.1 数据传输的控制机制

- ◆ 根据控制数据传输方法，总线分为：同步和异步总线
- ◆ 同步总线：靠时钟来控制数据传输
- ◆ 优点：传输延迟低，提高系统时钟频率，提升工作带宽
- ◆ 缺点：一旦确定了系统时钟，在以后的系统扩充中，采用新的高速期间的模块仍然要在原有时钟频率下工作；若提高系统时钟频率，则系统中原有的速度慢的模块又无法工作

## 同步总线



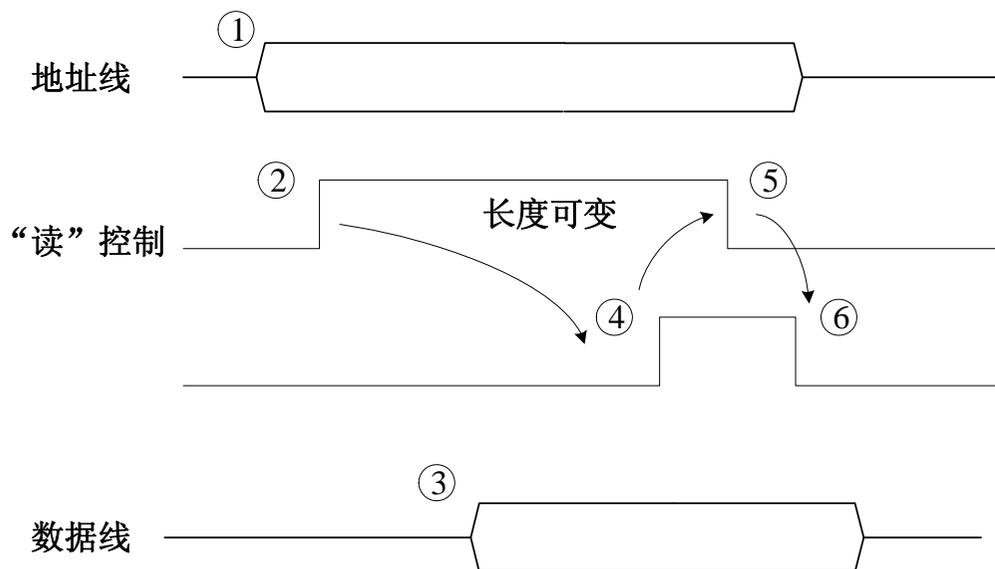
(a) 读操作



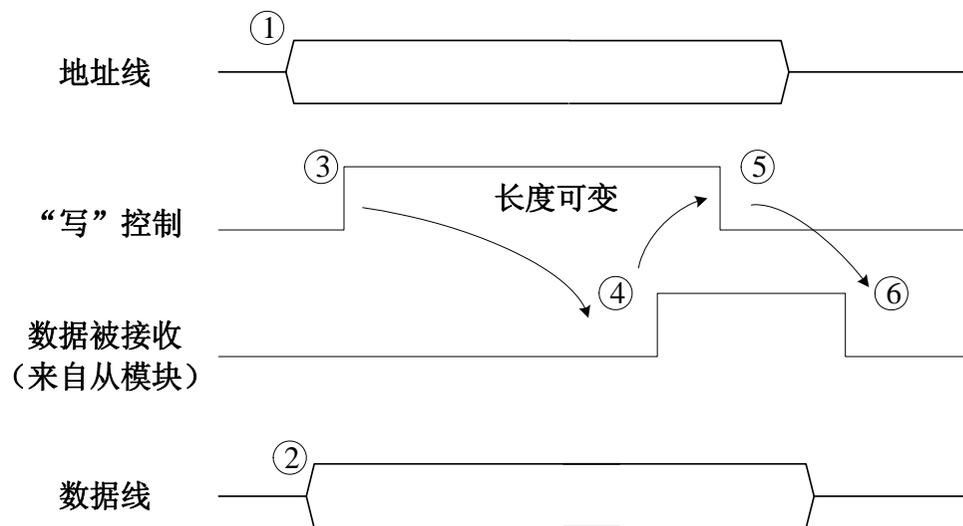
(b) 写操作

## 2.1.1 数据传输的控制机制

- ◆ 异步总线：靠握手信号来控制数据传输。
- ◆ 缺点：协议硬件复杂，每次交易开销大。
- ◆ 优点：对于不同工作速度的模块有很强的包容性，因此成为主流背板总线



(a) 异步读操作



(b) 异步写操作

# 2.1.2 VME总线

- ◆ **VME总线, Versamodel Eurocard**由Motorola公司1981年推出的第一代**32位**工业开放标准总线
- ◆ **VME64**,1995年推出新一代架构,加大了传输带宽,拓展了地址空间和方便了板卡插拔。它将**6U板**的数据线宽和地址范围扩展到了**64位**,给**3U板**提供了**32位**和**40位**地址模块,传输带宽达到了**80Mb/s**,增加了总线锁定周期,增加了第一插槽探测功能,加入了对热插拔的支持。
- ◆ **VME64x**是1997通过的新标准,它增加了一个**160管脚**连接器系列(按**5行**排列),还在**P1/J1**和**P2/J2**之间加入了一个**P0/J0**连接器,另外新增设了一个**3.3V**电源管脚。该系统新增的两个边缘总线循环则把数据速率提高到**160 Mb/s**。此外,还增加了**EMC前置面板**和**ESD功能**。

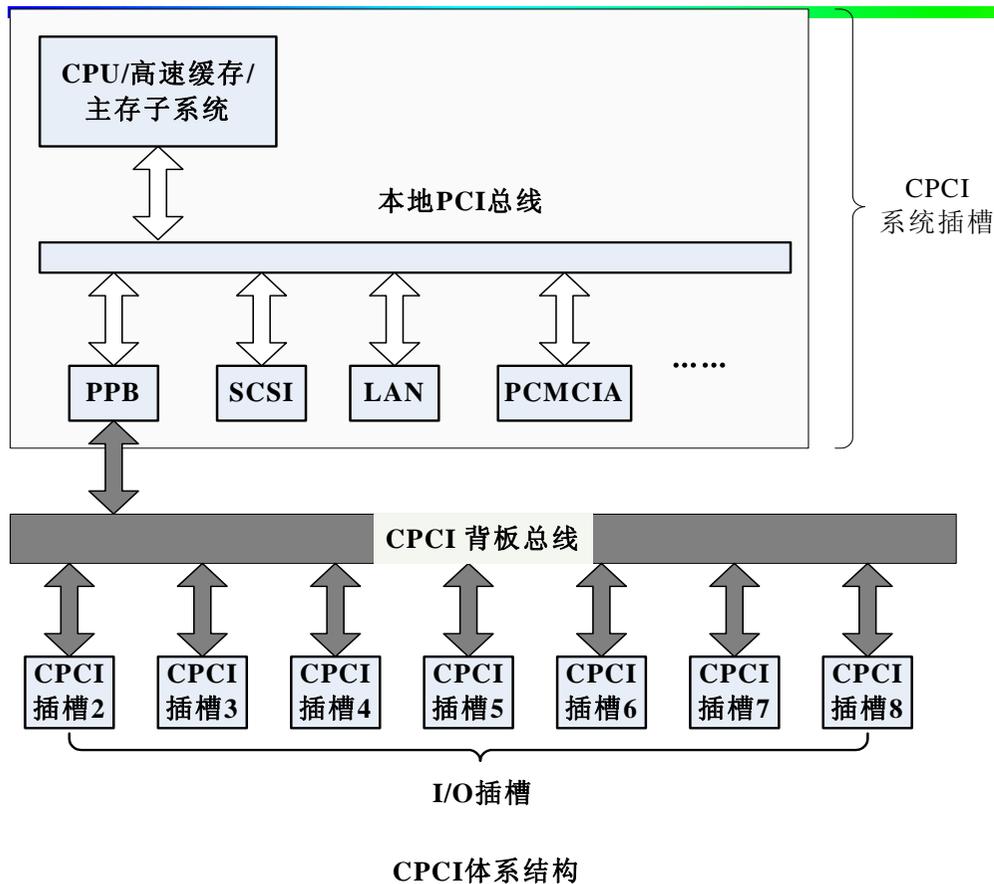
VMEbus基本特性

体系结构	主/从结构
传输机制	异步总线
寻址范围	16,24,32 (32根地址线)
数据宽度	8,16,24,32 (32根地址线)
中断能力	7级
多处理机能力	1~21个处理机
数据传输率	40 M byte/s
机械尺寸	6U, 3U
最大插槽数	21槽

背板总线常用的模块尺寸

规格	电路板尺寸	前面板高度	连接器数目
3U (单高)	100mm x 160mm	~128mm	1
6U (双高)	233mm x 160mm	~262mm	2
9U (三高)	367mm x 400mm	~395mm	3

## 2.1.3 CPCI总线



CPCI基本特性

体系结构	主/从结构
传输机制	地址/数据复用同步总线
寻址范围	32bit/64bit
数据宽度	32bit/64bit
中断能力	7级
多处理机能力	1~21个处理机
数据传输率	133Mbyte/s
机械尺寸	6U, 3U
最大插槽数	8槽(1个系统槽, 7个I/O插槽)

- ◆ CPCI背板总线是从PC台式机的PCI总线发展而来。
- ◆ 第1插槽为系统插槽，采用集中式仲裁方式和自动配置地址空间的机制，不能用于对等方式的多处理机系统

# VME总线&CompactPCI

	VME总线	CompactPCI总线
历史	1981	九十年代中期
标准组织	VITA	PICMG
板型	3U/6U/9U	3U/6U
模式	异步	同步
位宽	32/64位	32/64位
带宽	D64: 80MB/s	64位: 264MB/s
	2eSST: 320MB/s	532MB/s
仲裁	集中式仲裁/4优先级	集中式仲裁
中断	集中&分布/7优先级	集中/4级
扩展性	21槽	8槽/段
应用	国防/工业自动化/核工业	电信/工业自动化/国防

## 2.1.4 I/O接口模块

---

### ◆ I/O接口类型:

- ❖ 并行数字I/O接口
- ❖ 串行数字I/O接口
- ❖ 模拟信号I/O接口: 模数和数模转换
- ❖ 时间信号I/O接口: 时间数字转换器TDC

### ◆ I/O访问方式

- ❖ I/O映射方式: CPU寻址范围小, 存储器空间有限
- ❖ 存储器映射方式: VME I/O
- ❖ DMA方式: 高速传送大量数据的外部设备

## 2.1.4 I/O接口模块用途

### ◆ 模拟量输入模块A/D

采集来自设备上连续变化的物理量，如温度、压力、流量、电流、电压等。输入信号为电压信号（如0-10V）或电流信号（4-20mA）。AD通过端子板、连接电缆、信号调理模块将输入的模拟信号转换为数字信号送入计算机。

### ◆ 模拟量输出模块D/A

来自计算机的数字信号经过D/A转换通过信号调理模块、连接电缆和接线端子板送到设备控制器等执行机构或直接送到设备上。一般输出为0-10V或4-20mA的连续直流信号。

## 2.1.4 I/O接口模块用途

---

### ◆ 数字量输入模块DI (Digital Input)

常用来采集被控设备的运行状态，比如设备的开关状态等。一般是触点信号或电平信号，比如TTL电平或24V直流信号等。

### ◆ 数字量输出模块DO(Digital Output)

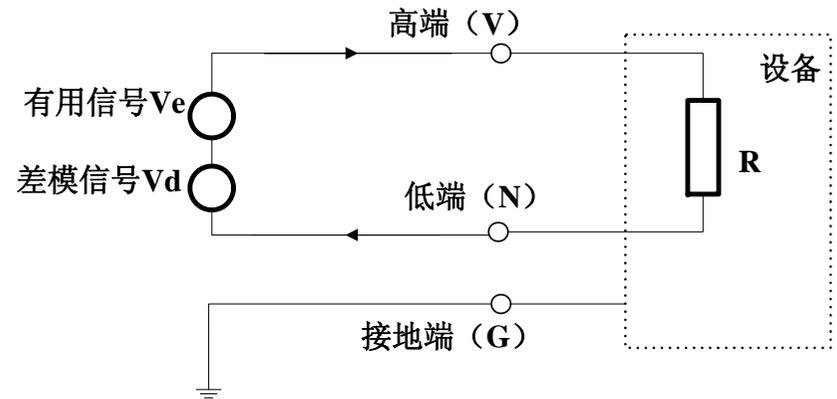
数字量输出信号用于向执行机构或被控设备送出操作信号，执行开关机，或者做各种运动的操作。一般是触点信号或电平信号，比如TTL电平或24V直流信号等。

## 2.2 电磁兼容与接地

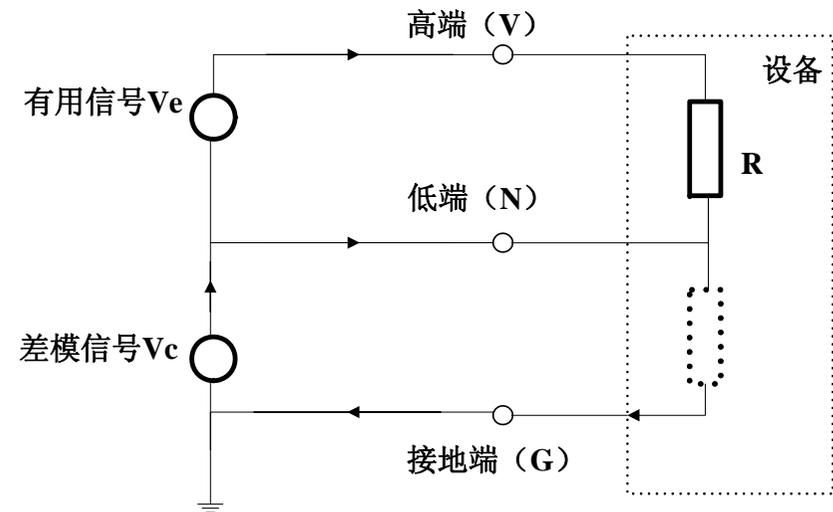
- ◆ 控制系统中前端控制设备一般放置在电源设备附近，工作在强电磁干扰环境下，因此，控制硬件设备需要采取隔离、屏蔽、合理接地等抗干扰措施，减少强电磁场对控制系统弱电信号的干扰，使系统能够正常工作。
- ◆ 控制系统中的电磁干扰和抗干扰技术：
  - ❖ 空间电磁干扰，主要依靠对机箱和电缆的屏蔽来克服
  - ❖ 来自供电系统的干扰，主要采用隔离变压器、交流电源滤波来抑制。
  - ❖ 外部噪声通过导线串扰产生的差模干扰，主要采用信号滤波的方法。例如在输入端的上游安置滤波器滤除噪声
  - ❖ 由地回路而产生的共模干扰，主要采用信号地输入浮置、差分输入方法克服

# 差模干扰和共模干扰产生原因

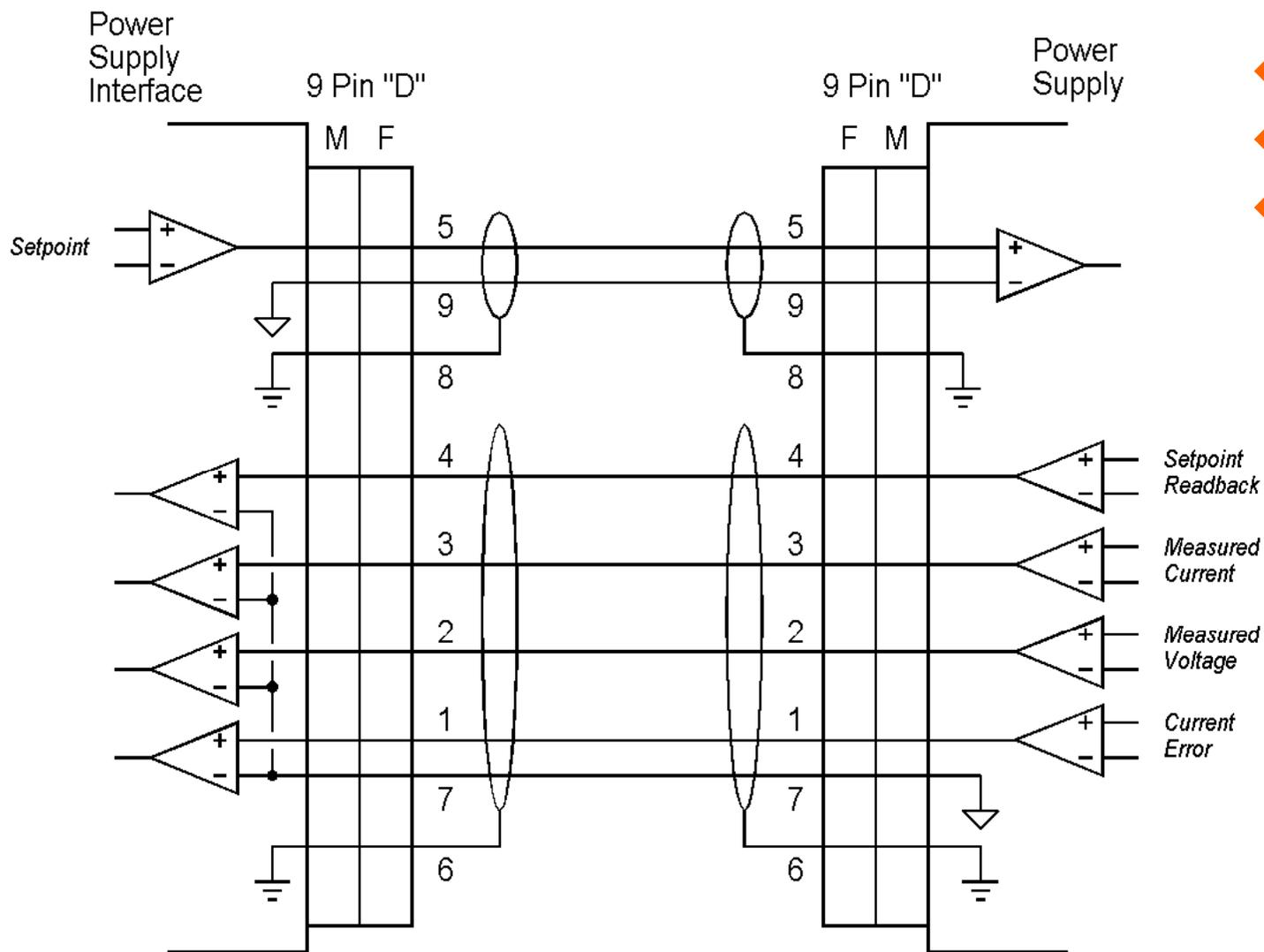
- ◆ 差模干扰: 噪声电压所产生的干扰电流与所期望的(有用的)电流相串联的干扰 (**differential mode**)
- ◆ 共模 (**common mode**) 电流是指同方向加在高端和低端两条线上, 并通过地线返回的电流。
- ◆ 共模干扰只有在**3根**输入导线的情况下才会产生



差模干扰



# 差分输入的实例-电源接口电缆连接

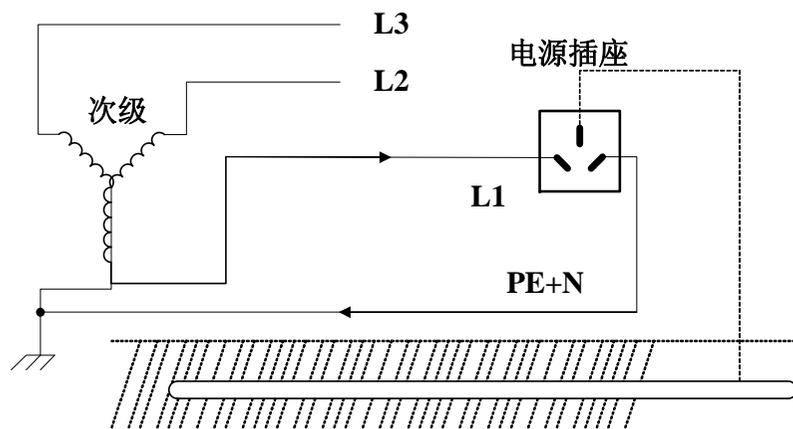


- ◆ 差分输入
- ◆ 信号负端浮地
- ◆ 屏蔽层接入大地

# 控制系统的接地

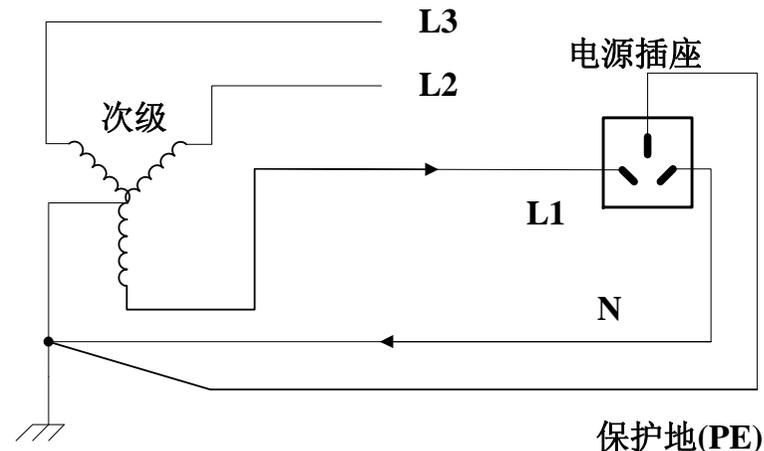
- ◆ 电力系统的接地
- ◆ 中线接地
- ◆ 保护地:电气设备和电子学设备普遍采用导电的金属作为结构和包封材料,如果火线与金属机箱绝缘不良,漏电就会通过杂散阻抗使得机箱带有较高的电位,如果人体触摸机箱,就会造成触电事故,为此将金属机箱接地,这个地称为保护地(PE, protective earth)
- ◆ 三相五线制:
  - ❖ 中线N和保护地线PE是各自独立和相互绝缘的导体,只是在供电系统的源端(电力变压器次级)处才接在一起并接到大地电极。因为保护地与中线完全独立,在正常工作状态下,保护地线不会带任何电流,十分“干净”。
  - ❖ 控制系统使用供电系统提供的现成的保护地作为公共地线网络。
- ◆ 控制系统为了更好地防止从电源引入的各种干扰,采用隔离变压器和不间断电源(UPS)供电,在UPS接入之后,保护地的接法与不用UPS时的接法没有任何不同。

# 控制系统的供电接地



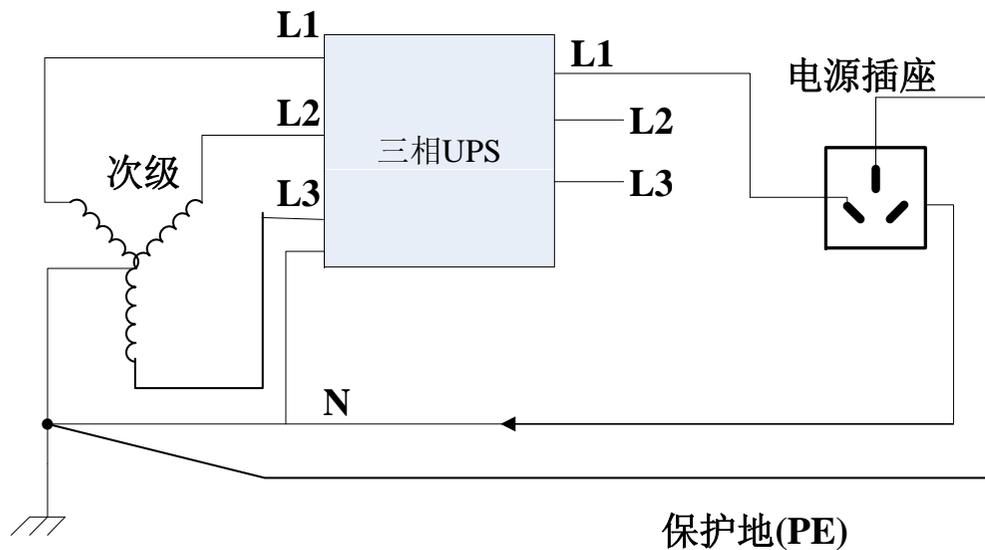
金属管道

(a) 旧的三相四线制



保护地(PE)

(b) 三相五线制



保护地(PE)

(c) 三相五线制的UPS供电

# 控制系统的接地

- ◆ 电路地提供设备中的电子电路模块的电位参考点，在许多情况下，电路地同时被用作信号地、信号返回通路和直流电源的返回通路。
- ◆ 设备地通常就是设备的金属机箱，是整个设备的电位参考点，又称为机箱地。系统中的各个设备地都连接到保护地线，同时也起到防静电接地的作用。
- ◆ 在通常情况下，设备之间多采用结构简单、便于实施的多点接地方式。在多点接地方式下，地回路是引进干扰的重要原因。为了切断地回路，最有效的方法是信号隔离。在分布式计算机系统中，数据通信采用光纤网，是最实用的信号隔离方法。
- ◆ 设备之间的单点接地方式，固然能有效地切断地回路，但是实施起来比较复杂，在强的地回路干扰的环境下，当无法采用信号隔离时，对于噪声敏感的设备之间，可以采用专门的地线和信号地单点接地方式。

## 2.3 网络和数据通讯

- ◆ 网络在控制系统中的应用分为两个层次：
- ◆ 一个是用于操作台计算机和前端控制计算机之间交换信息的网络，称为信息层网络，广泛使用的以太网协议和**TCP/IP**协议。
- ◆ 另一个网络层次是用于控制现场的元件或设备的网络，称为控制层网络，即各类现场总线。

- 各种网络拓扑及其特点

- **ISO/OSI**网络参考模型

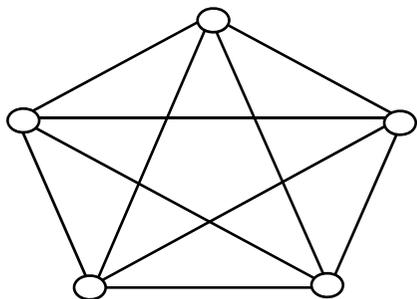
- 传输介质

- 数据通讯方式

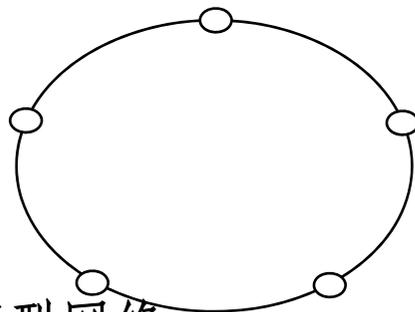
- 经典的数据通信协议：  
**RS-232**和**RS-485**

- 现场总线技术

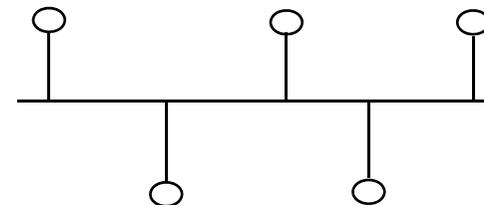
## 2.3.1 各种网络拓扑及其特点



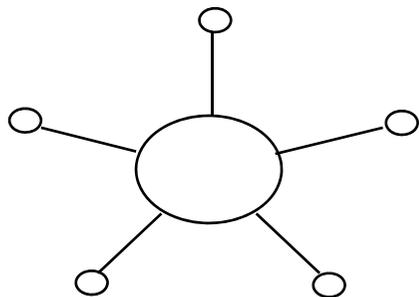
物理的点对点连接  
布线多，通信效率高



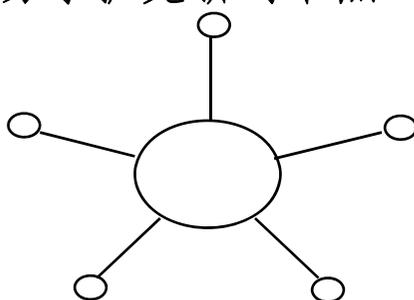
环型网络  
布线多，通信效率底  
不易于扩充新的节点



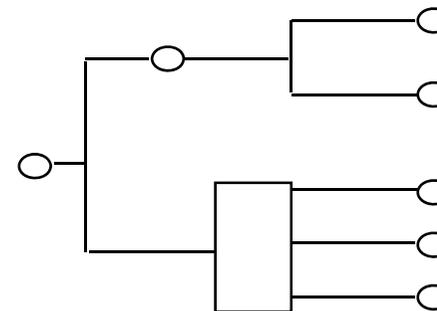
总线型网络  
布线简单易于扩充新的节点  
故障诊断和故障隔离困难



星型连接-主从方式



星型网络-交换方式  
可靠性强



树型（混合型）网络  
常用

## 2.3.2 ISO/OSI网络参考模型

- ◆ 国际标准化组织（ISO）提出一个网络体系的协议，称为开放系统互连（ISO/OSI）参考模型

### OSI 参考模型各层的功能

7	应用层（application）	向 OSI 模型之外的应用程序提供服务
6	表示层（presentation）	处理和转换数据格式（数据表示）
5	会话层（session）	建立、管理和终止与应用程序之间的会话
4	传送层（transport）	将信息分组并编号（端口号，即处理该信息的进程地址）
3	网络层（network）	选择路由，管理网络（IP）地址，进行流量控制和网络互连等
2	数据链路层（data link）	<ul style="list-style-type: none"><li>- 建立节点间的数据链路，传输按格式（消息帧）组织的位码</li><li>- 检测和校正物理链路上的差错</li></ul>
1	物理层（physical）	<ul style="list-style-type: none"><li>- 建立、维持和拆除与电、机械、功能和过程的有关的物理链路</li><li>- 在链路上传输没有结构的位流（bit stream）及故障检测等</li></ul>

## 2.3.2 ISO/OSI网络参考模型

◆ 7层OSI模型分为两部分:

❖ 第1、2层为数据通信协议;第3至7层为网络协议。

应用层	应用层	HTTP 协议	FTP 协议	Email	DNS	各种 应用程序
表示层						
会话层						
传输层	传输层	TCP (传输控制协议)		UDP (用户数据报协议)		
网络层	网际层	IP (网际协议)				
数据链路层	网络接口层	以太网	令牌网	FDDI	.....	
物理层		同轴电缆	双绞线	光缆	无线	

OSI 模型层

TCP/IP 层

## 2.3.3 传输介质

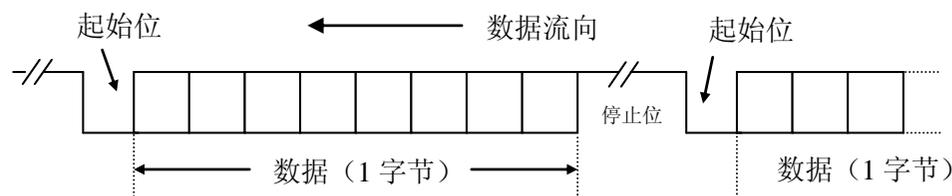
- ◆ 数据传输常用的介质为双绞线、同轴电缆和光纤。
- ◆ 双绞线（**twisted pair**）是一种廉价、连接方便、适合用于差分传输并且抗电磁干扰良好的传输介质，经常用于传输距离不远（**100米量级**）的计算机数据通信。在网络中最常用的是“**5类无屏蔽双绞电缆**”，它包括**4对双绞线**。
- ◆ 同轴（**coaxial**）电缆的传输损耗小、传输距离远、频带宽、抗干扰性能好。目前已被双绞线所取代。
- ◆ 采用玻璃或塑料光导纤维的光缆，是性能最好的传输介质。

# 数据传输方式

- ◆ 同步传输方式使信号的发送方和信号的接收方在传输数据时使用共同的时钟。
- ◆ 异步传输中，发送方和接收方在传输数据时使用各自的时钟，收、发的时间关系很容易漂移而导致传输错误。



(a) 同步传输



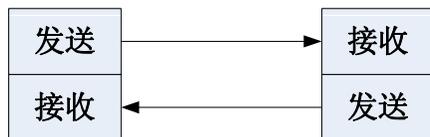
(b) 异步传输

同步传输和异步传输的代码结构



(a) 单工通信

(b) 半双工通信



(c) 全双工通信

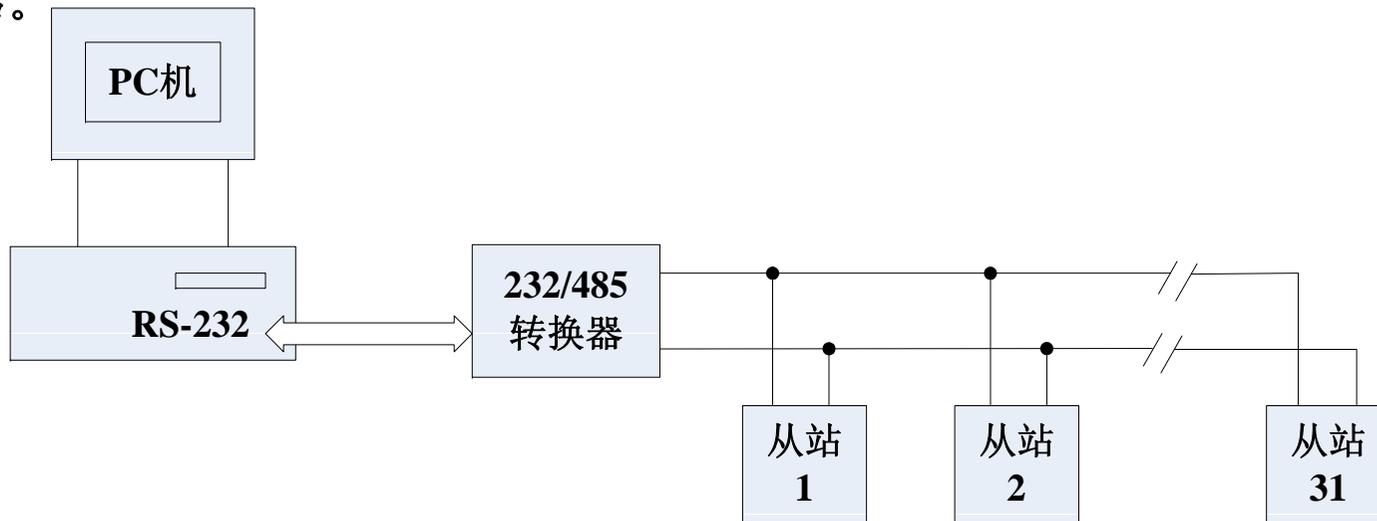
- ◆ 单工通信：数据只能沿一个方向传输的方式
- ◆ 半双工通信：数据可以双向交替传输的方式
- ◆ 全双工通信：双方使用两条通路同时进行双向传输的方式

## 2.3.4 经典的数据通信协议

- ◆ 美国电子工业协会（**EIA**）在**1962**年制定的**RS-232**标准是最广泛使用的数据通信协议，用于点到点的串行异步通信。
- ◆ **RS-232**协议只规定了接口的电气和机械特性
- ◆ **RS-232**通信技术不断地改进，早期常用的传输速率仅为**4800bps**或**9600bps**，而目前最大通信速率已可达到**250Kbps**，但对于可靠的应用，通信速率仍限制在**20Kbps**。
- ◆ **RS-232**标准的传输距离为**50 英尺（15米）**
- ◆ 在控制系统中，往往在上游采用具有多个**RS-232**端口的模块，分别控制下游的若干设备，形成主从方式的星型连接。

## 2.3.4 经典的数据通信协议

- ◆ **RS-485**是多点连接方式、平衡（差分）传输的数据通信协议。
- ◆ 共模抑制能力强，抗干扰性能好，对外部产生的干扰也小
- ◆ **RS-485**驱动（发送）芯片具有3态输出，可以用**enable**控制输出处在高阻态，利用一对线进行多点的双向传输，对于100米的传输线，最大传输速率为1Mbit/s。
- ◆ **RS-485**只定义了电气特性，并没有规定连接器和电缆标准。
- ◆ 典型的**RS-485**应用是使用1对双绞线组成半双工通信系统
- ◆ 在过程控制中，可以用**RS-485**以主从方式组成小规模的控制系統，采用半双工通信方式，其中1个节点用作主站，其他31个节点为连接被控设备的从站。



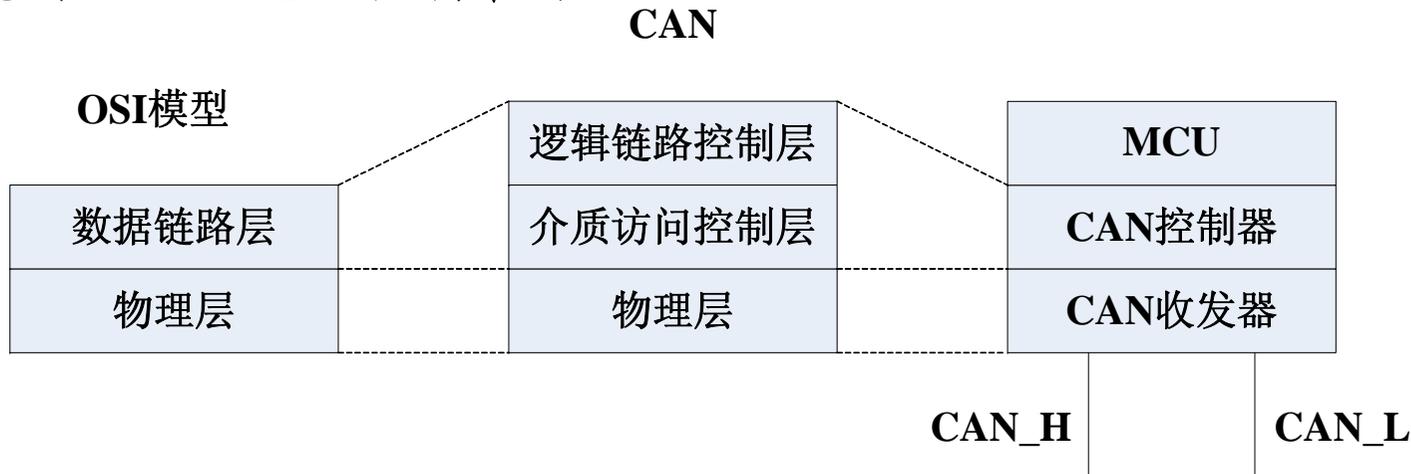
## 2.3.5 现场总线

---

- ◆ 现场总线（**fieldbus**）是用来将控制器与分布在现场的各种类型I/O设备或I/O元件相连接的网络。
- ◆ 现场总线主要采用总线拓扑结构，具有电缆开销小、可靠性高和灵活性强的优点，在工业控制中得到广泛的应用。
- ◆ 常用的现场总线
  - ❖ **CANbus/DeviceNet**、**Profibus**、**WorldFIP**、**LonWorks**、**FL-net**和**ControlNet**

# CANbus

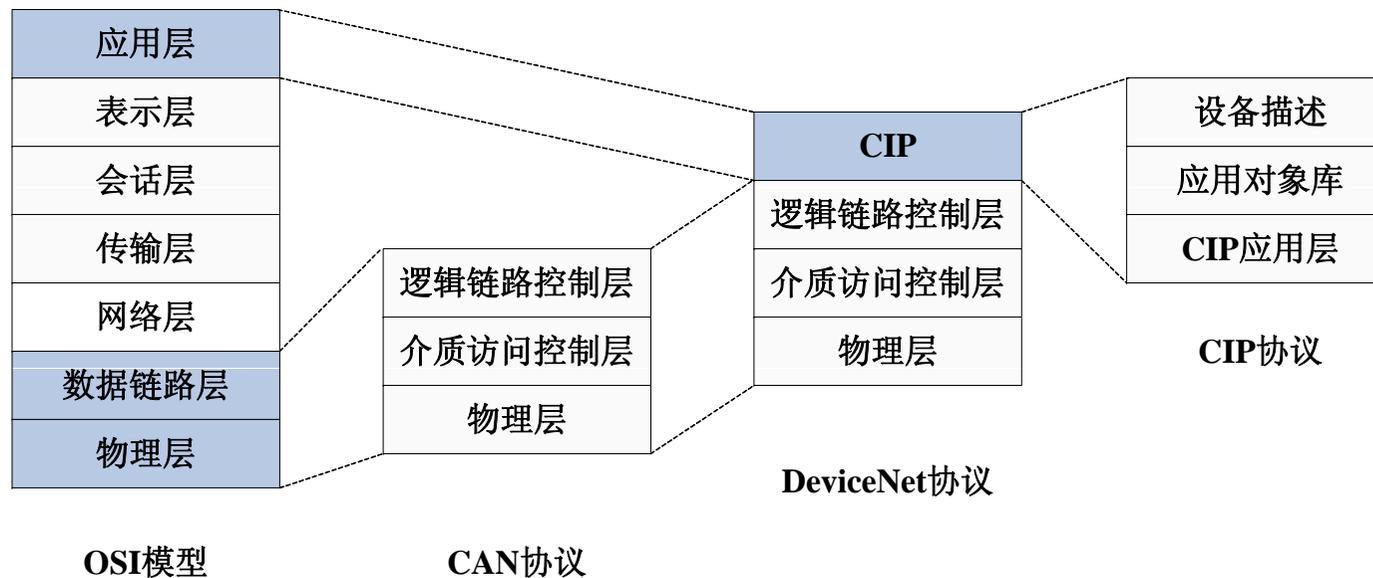
- ◆ **CAN (Controller Area Network)** 的基本协议层只包括物理层和数据链路层。
- ◆ 数据链路层又分为介质访问控制层和逻辑链路控制层两部分
- ◆ **CAN**的通信接口由**CAN收发器** (对应物理层) 和**CAN控制器** (对应介质访问控制层和逻辑链路控制层) 组成。
- ◆ 总线通常采用双绞线作平衡式传输, 两根信号线分别称为**CAN\_H**和**CAN\_L**。
- ◆ **CAN**的传输距离较短, 通常只在**500米**以内, 支持**32 ~ 64**个节点。
- ◆ 当数据传输速率**1Mbps**时, 节点间最大距离为**40m**, 当数据传输速率**500bps**时, 节点间最大距离约为**100m**。
- ◆ 德国**BESSY**电源控制采用



# DeviceNet

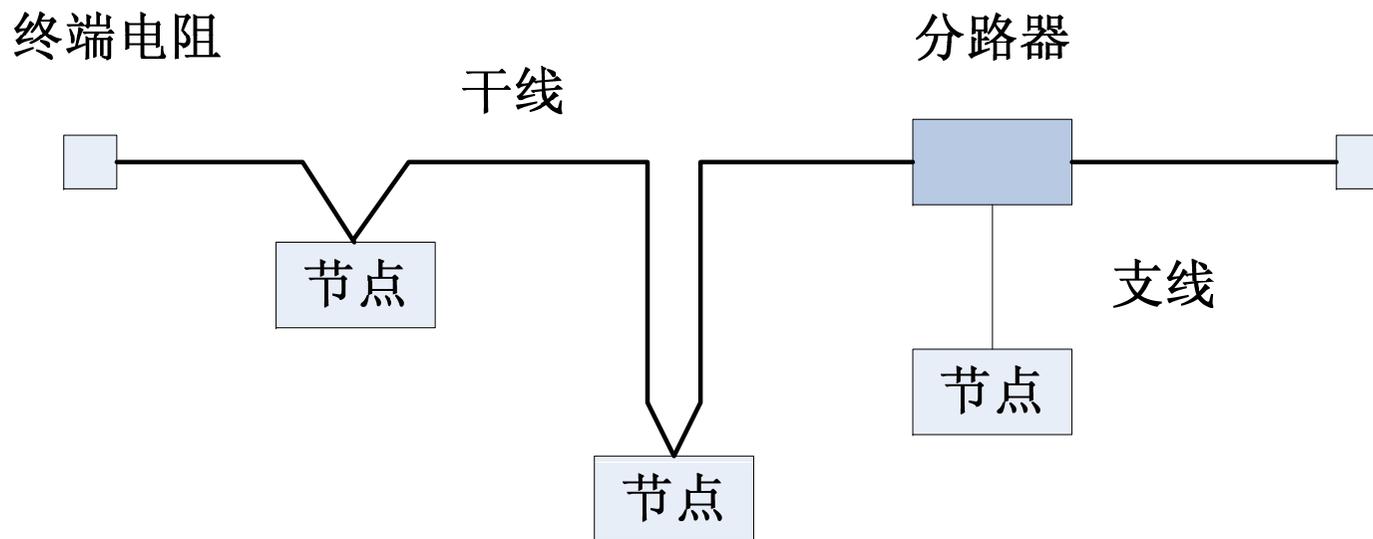
◆ 对**CAN**的扩展包括对物理层协议的补充以及增加应用层协议。应用层协议采用**CIP**协议（**Common Industrial Protocol**），**CIP**协议本身可以分为以下层次：

- **CIP**应用层，用于数据管理、消息服务和路由管理等等
- 应用对象库
- 设备描述（**Device Profile**）



# DeviceNet

- ◆ DeviceNet的电缆由5根线组成
- ◆ 1对双绞线用于通信
- ◆ 1对双绞线用于提供电源，还有1根屏蔽线。
- ◆ DeviceNet的网络分为干线（Trunk Line）和支线（Drop Line），支线通过分支器（Tap）可以很方便地从干线上引出



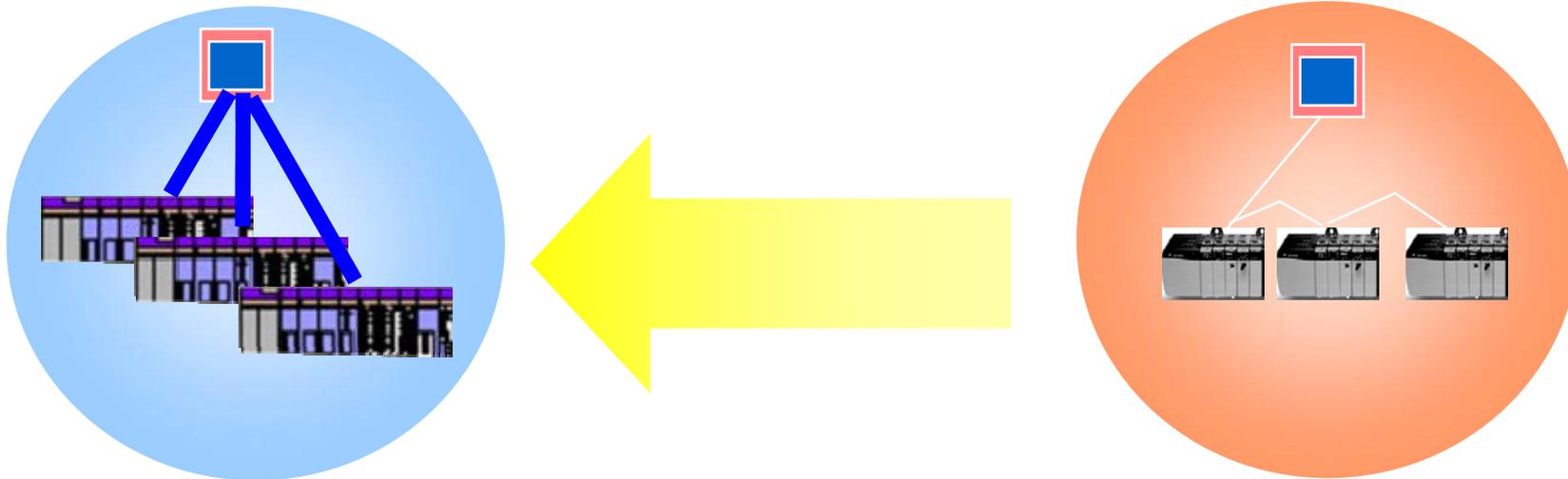
DeviceNet网络基本组成

# FL-net

- ◆ **FL-net**是由日本电机工业会、制造科学技术中心、日本自动车工业会、日本工作机工业会、日本机器人工业会和日本电气计测器工业会等合力倡导开发的日本**PLC**网络标准
- ◆ **FL-net**是以**Ethernet**为其物理层，采用总线的拓扑结构，通信站的管理为无主方式，令牌总线方式的通信权控制，在其协议栈的高层为**UDP/IP+**独自循环通信协议，保证了通信的实时性，**FI-net**的数据传输速率为**10 Mbps**。
- ◆ **FI-net**具有总线冗余、站点冗余、网络通讯单元的故障自动校验等功能，保证通信的可靠性。通信电缆可采用双绞线、光纤或同轴电缆，最大传输距离**10km**。
- ◆ 在**FL-net**总线上可以直接挂接**PC**机，**PLC**和独立的**I/O**模块，触摸屏以及智能传感器等设备，因此系统的建造和扩充有较大灵活性。
- ◆ **FA-M3**提供图形的**WideField2**编程软件，简单易用，提供模块化程序功能、本地设备编号、宏、设备结构说明等新的编程功能，减少用户开发成本，提供了程序的可再利用性。
- ◆ **J-Parc**、**SSRF**和**CSNS**控制系统采用

# 比较: FL-net VS. Controlnet

- 可使用即存 Ether LAN
- Max 255 站、扩展距离无限制。
- 使用Router 方式，1站故障不影响其他站工作。且可随时随地加入脱离
- 开放网络，为与PC及与其他PLC连接提供可能
- 简单设定无需编程
- 可通过模块对CPU编程监视。（FA-M3独有）



## 2.4 时间信息的获取与传输

---

- ◆ 时间是控制系统中的一个重要的物理量。时间信息是与特定的事件相关联的，在控制系统中，时间信息处理主要归结为3个方面：
  - ❖ 时间信息的测量。即测量两个特定事件发生的时间的间隔，通常采用时间/数字转换器实现测量。
  - ❖ 通过定时系统，将定时信号传播给各个需要时间触发的电子学设备。
  - ❖ 向分布式计算机系统的各个节点提供实时钟，用以建立各节点间的时间同步关系。

# 时间系统的分类

传播时间信息的系统可以分为定时系统和时间同步系统。

## ◆ 定时系统

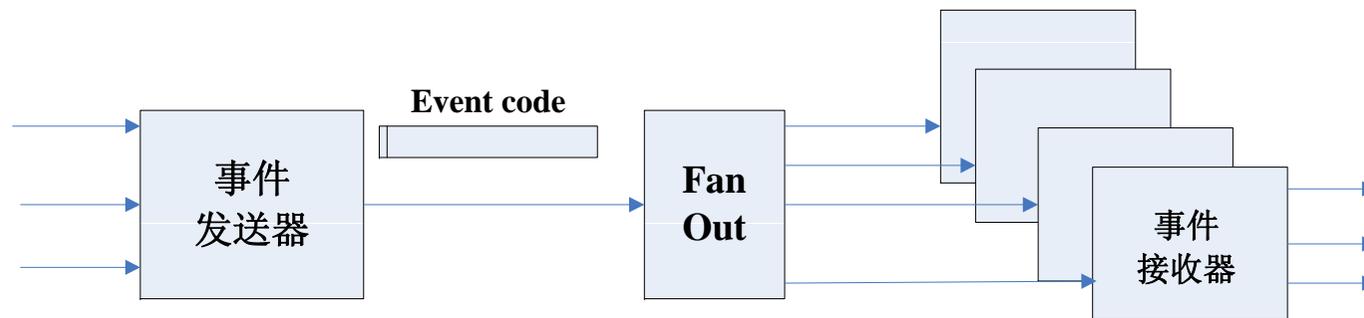
- ◆ 定时系统所传递的时间信息是基于相对时间的，即基于某个时间参考点的的时间值。
- ◆ 在分布式控制系统中，若干分布在不同位置的设备，需要通过定时系统从主时钟获得相对时间信号，用于触发（或延时触发）执行特定的控制（或测量）。
- ◆ 这种系统级的定时/触发系统所要求的时间精度因设备的功能而异，可以从ns量级到ms量级不等

## ◆ 时间同步系统

- ◆ 时间同步系统所传递的时间信息是基于绝对时间的，即基于日期（年月日时分秒）的。
- ◆ 分布式控制系统中，不同计算机节点中运行的进程之间往往需要保证同步/互斥关系，所进行的测量或控制也需要满足时间因果关系。
- ◆ 需要通过网络通信建立时间同步系统，由主时钟节点向所有其它节点传送称为时间印记（time stamp）的短数据包，提供统一的参考时间，用来校正各节点的本地时钟，实现时间同步。

# 定时系统

- ◆ 随着高速数字传输技术的发展，使用数字信号来实现数字定时系统
- ◆ 国内外许多实验装置如SLS、Diamond、LCLS、SSRF、BEPCII使用EVG/EVR数字定时系统。
- ◆ 利用现成的Gigabit以太网物理层协议（即收发器件和光缆），通过事件发生器（Event Generator, EG）和事件接收器（event Receiver, ER）建立联系。发送方（EG）将输入的触发定时信号转换为16bit事件码（event code）发送出去，接收方（ER）从事件码中提取并输出定时信号。
- ◆ 事件发生器输入的触发信号可以来自硬件触发，也可以是软件触发。
- ◆ 各种设备使用同样的事件接收器，它们根据事件码所指定的内容（延迟时间和脉冲宽度等等）输出各种定时信号。



数字定时系统

# 时间同步系统

- ◆ 时间同步系统：由提供标准参考时间的主时钟节点（**Master**），通过网络广播方式，向所有其它从节点（**slave**）以固定的时间间隔发送时间印记（**time stamp**），各节点收到时间印记后，对本地时钟进行校正。
- ◆ 在网络和分布式系统中常采用时间同步协议：
  - ❖ **NTP/SNTP**协议（用于广域网和局域网）
  - ❖ **GPS**（全球定位系统，无线传输）
  - ❖ **PTP**协议（**Precision Time Protocol**），即**IEEE1588**协议

时间同步协议的比较

	<b>SNTP/NTP</b>	<b>GPS</b>	<b>IEEE1588</b>
分布范围	广域/局域网	全球	几个子网范围
通讯方式	互联网	卫星	局域网
精度	几个ms	<μs	<μs
管理方式	配置	五	自组织
硬件	无	接收机	有或无

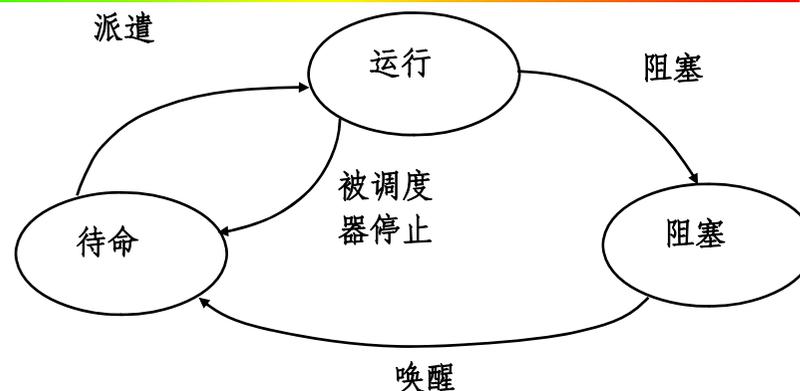
## 2.5 实时多任务系统

---

- ◆ 实时计算机系统的定义是：能够在确定的时间内运行其功能并对外部的异步事件作出响应的计算机系统。
- ◆ 现代控制系统广泛使用实时操作系统（**Real Time Operating System, RTOS**）构建实时多任务系统。
- ◆ 分为2部分：
  - ❖ 实时操作系统内核
  - ❖ 输入输出I/O部分

# 2.5 实时多任务系统

- ◆ 分布式控制系统中的各个节点普遍使用单处理机系统。单处理机系统在实时操作系统调度的下，可以使若干任务并发地运行，构成所谓多任务（multitasking）系统
- ◆ 任务及任务切换
  - ❖ 任务就是运行中的程序，包含程序和相应的数据。
  - ❖ 每个任务在系统中有一个任务控制块TCB
  - ❖ 任务切换（task switch）是指一个任务停止运行，而另一个任务开始运行。
- ◆ 实时任务调度
  - ❖ 周期性任务
  - ❖ 轮转方式任务调度
  - ❖ 基于优先级的抢先调度
    - 实时操作系统中，任务的优先级可达256级（例如0级最高，255级最低）



任务的状态转移

任务控制块TCB

任务状态
任务优先级
事件或资源计数
程序起始地址
初始化堆栈指针内容
程序计数器内容
状态寄存器内容
堆栈指针寄存器内容
各个通用寄存器内容

# 任务间通信机制

- ◆ 任务的同步和互斥
- ◆ 任务的同步：指两个或两个以上的任务需要按一定的时序协同工作
- ◆ 任务的互斥：对计算机的I/O设备、存储器、CPU时间以及时钟等共享资源的竞争
  - ❖ 信号灯（**Semaphore**），用于任务之间的同步和互斥。
  - ❖ 共享存储区（**Shared Memory**），用于任务之间的互斥
  - ❖ 队列就是一个线性表，所有数据从表的一端写入，从表的另一端被读出或删除
  - ❖ 管道类似于队列，它以**FIFO**方式工作，主要用来传递以字符为单位的文件。被看作一个虚拟的I/O设备。

## 2.6 控制软件设计

- ◆ 软件是计算机系统的灵魂。在加速器控制系统中，控制软件分为：
  - ❖ 设备应用软件：在实时多任务操作系统的管理下，最主要的应用软件模块有两种，任务模块和设备驱动器模块
  - ❖ 物理应用软件
- ◆ 控制软件设计方法
  - ❖ 结构化的分析和设计方法（**SA/SD**），又称面向功能的设计方法。
  - ❖ 面向对象的分析和设计方法（**OOA/OOD**）

## 2.6.1 结构化的设计

---

### ◆ 自顶向下（**top-down**）的方法

❖ 优点：易实现，适合一个特定的应用

❖ 缺点：不易扩充

### ◆ 自底向上（**bottom-up**）的方法

❖ 优点：由小到大地组织概念，易于扩充。

❖ 缺点：开始时不了解上层的需要，又往往导致底层开发工作量大

### ◆ 实用中，通常采用自顶向下与自底向上相结合的方式。

❖ 对于控制系统，可以先行开发（或选定）特殊的I/O设备及与之相应的驱动器软件，实现各种底层功能并由此获得对于系统可行性的充分知识，然后开始自顶向下的设计。

## 2.6.2 面向对象的设计方法

- ◆ 面向对象（**Object Oriented**）的分析和设计方法（**OOA/OOD**）是当前最为流行的软件设计方法，在控制系统软件设计中得到了广泛的应用。
- ◆ 面向对象就是把一个系统看作是许多分立的对象（**object**）的集合，对象是具有明确边界和意义的、具体的或抽象的事物。
- ◆ 对象定义为封装了属性和行为（操作）的实体。
- ◆ 属性就是描述对象状态的数据结构
- ◆ 具有同样属性（数据结构）和行为（操作）的对象，可抽象为类（**class**）。
- ◆ 类是对象的抽象，而对象则是类的实例（**instance**）。
- ◆ 对象通过消息传递信息，消息可以激活执行指定的操作。

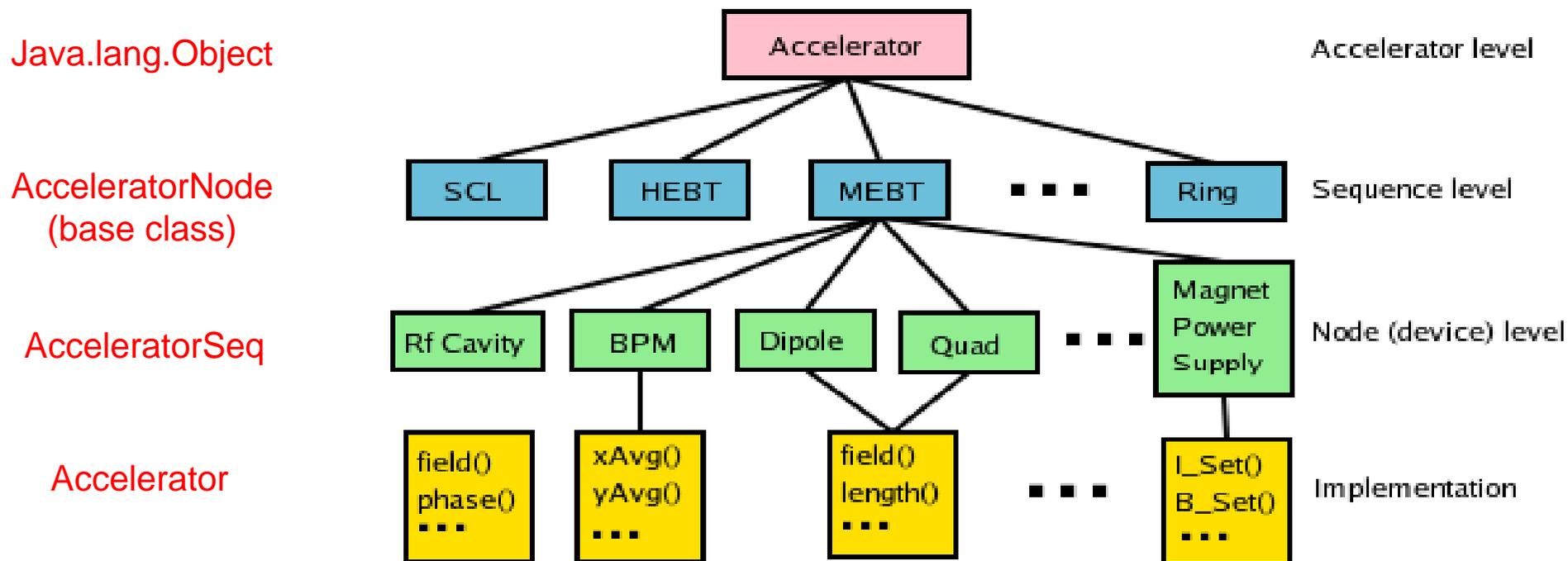
## 2.6.2 面向对象的建模

- ◆ 统一建模语言（**Unified Modeling Language, UML**）是当前流行的用于面向对象的分析及设计的符号标识系统。用于软件系统从需求分析直到详细设计的全过程，在各类系统工程设计中都得到广泛的应用。
- ◆ 一般的设计步骤如下：
  - ❖ 根据需求对用例建模。划分子系统，分析和定义系统的用例图，描述对系统感兴趣的角色及其对用例的各种功能的要求。
  - ❖ 通过分析，建立可以被广泛使用的类，定义它们的属性、方法、关系和行为。
  - ❖ 将分析所建立的类具体化，产生结构模型。确立类的对象，包括它们的数据结构、算法、操作和消息传递；定义类的层次结构。
  - ❖ 建立系统的行为模型，作出它们的活动图、状态图和交互图。
  - ❖ 建立系统的实现模型，全面描述系统的构件配置和软硬件体系结构。
- ◆ 对于控制系统/实时系统的建模，除了上述一般的考虑之外，还需要了解系统的时间需求，需要描述系统中的并发、通信、同步以及异步事件处理等各项机制，对于分布式系统，还需要描述系统的分布。

## 2.6.2 面向对象的建模实例-XAL

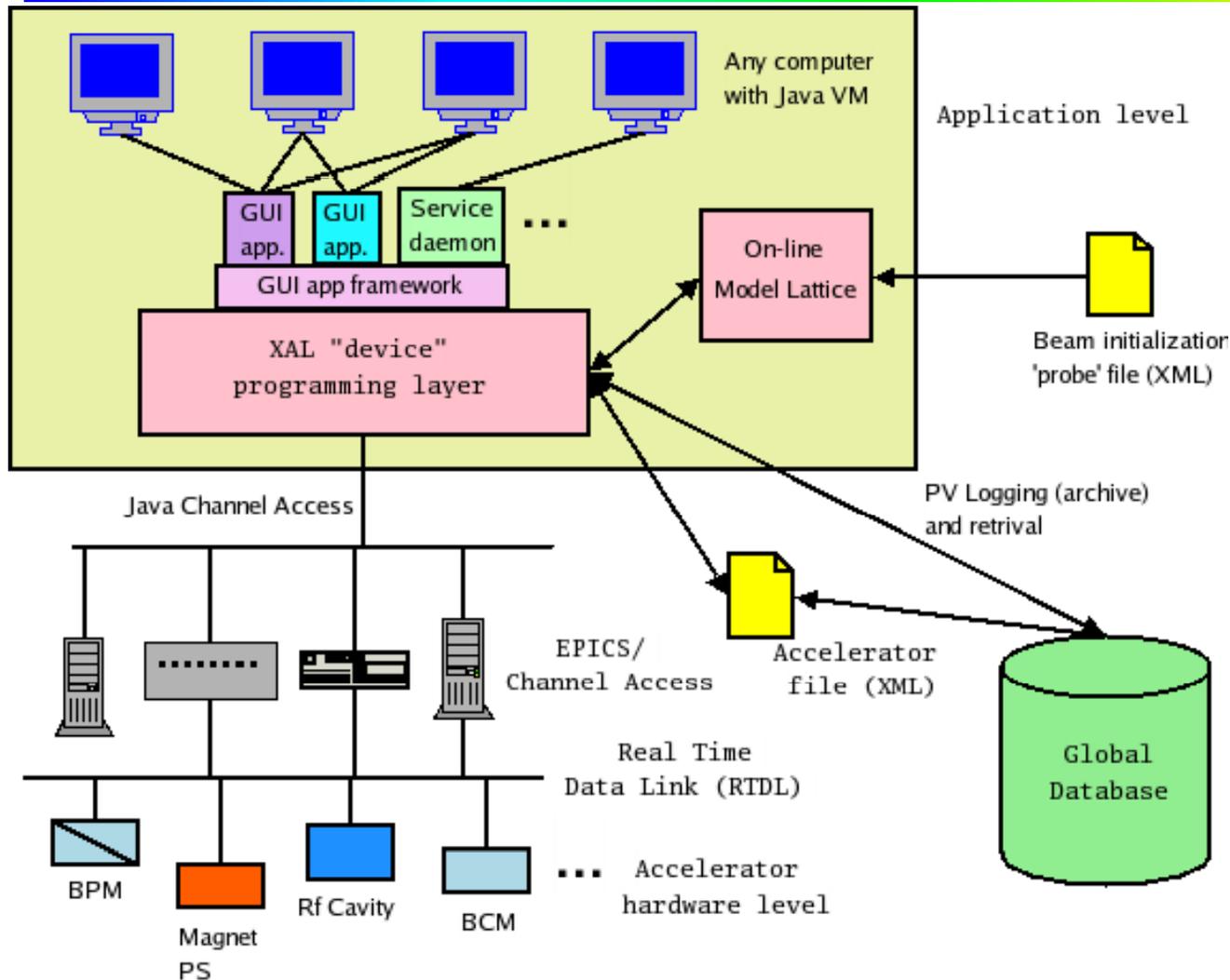
- ◆ 物理调束应用软件**XAL**
- ◆ 国际上成熟的大型在线加速器应用软件不多，大多是各大加速器实验室根据自己的情况零星地开发了一些特定的应用程序。
- ◆ 美国**ORNL**实验室为**SNS**项目所开发的**XAL**软件包则具有比较完备的功能和比较容易扩展的良好框架结构
- ◆ **XAL**采用面向对象的设计方法，它从加速器物理的角度出发，基于**Java**语言对组成加速器的元器件如磁铁、电源、束流测量探头（如**BPM**等）、射频腔等数据结构进行软件建模。
- ◆ **Provide on-line Model**
- ◆ **Virtual Accelerator**

# 加速器物理调束应用软件XAL



加速器的类定义层次结构

# XAL应用框架与EPICS控制系统的关系



- ◆ 数据库作用
- ◆ 用于映射组成加速器的元器件如磁铁、束测元件、射频腔等数据结构和静态参数
- ◆ 建立束流光学结构和设计参数在数据库中的物理映射
- ◆ 实时数据历史数据库

## 2.7 控制系统可靠性

- ◆ 加速器控制系统要求计算机控制系统在内的所有组成部分都具有高可靠性，这样整个系统才有基本的可用性。
- ◆ 可靠性指标
  - ❖ **MTTF**平均失效时间（**Mean Time To Failure**），是指系统的预期平均寿命，常用来衡量不可维护系统（即必须连续工作的系统）的可靠性。
  - ❖ **MTTR**平均修理时间（**Mean Time To Repair**），即系统每次失效后的平均修复时间。
  - ❖ **MTBF**平均失效间隔（**Mean Time Between Failures**），是平均失效时间与平均修理时间之和（ $MTBF = MTTF + MTTR$ ）。用来衡量非连续可用系统（即出了故障允许短暂停机维修的系统）的可靠性。

# 可靠性的度量

## ◆ 对于不可维护系统

❖ 失效率 (failure rate)  
 $\lambda$  是系统预期寿命 (即  
MTTF) 的倒数:  $MTTF = 1/\lambda$

❖ 可用度 (availability)  
 $A(t)$  定义为系统在  $t$  时刻  
工作正常的概率:  $A = 1 - \lambda$

❖ 例如可用度为  $(1 - 10^{-6})$   
= 99.9999% 的连续工作系  
统, 其失效率为  $10^{-6}$  / 小  
时, 即平均每年不可用时间  
约为 31.5 秒/年。

## ◆ 非连续可用系统的可靠性度量:

❖ 对于非连续可用系统, 即允许短暫停  
机维修的系统, 可用度定义为

$$A = \frac{MTTF}{MTTF + MTTR} = \frac{MTTF}{MTBF}$$

❖ 系统的 **MTTF** 越长, 或者 **MTTR** 越短,  
系统的可用度就越高。

❖ 设法减少 **MTTR**, 可以有效地提高系  
统的可用性。

## ◆ 对于可以进行短暂维修的非连续可用系 统:

❖ 失效率  $\lambda = 1 - A \approx$  运行期间的修理次数 /  
总的运行时间

❖ **MTBF** =  $1/\lambda \approx$  总的运行时间 / 运行期间  
的修理次数

# 高可靠性系统的分类

- ◆ 按系统要求高可靠性的目的来分类，通常分为生命安全型系统和任务确保型系统。
- ◆ 生命安全型系统（**Safety-critical System**或**Life-critical system**，或可译为生命交关系统），要求确保人身生命安全，失效率在 $10^{-7}$ /小时至 $10^{-10}$ /小时之间。  
如：民航客运飞机和核电站
- ◆ 任务确保型系统（**Mission-critical System**，或可译为使命交关系统）的基本要求是要确保完成任务，失效率在 $10^{-4}$ /小时至 $10^{-6}$ /小时之间，典型的例子有：军事作战系统，电信系统，专用系统和大型控制系统。

- ◆ 按照对可用度的要求来分类，高可靠性系统可以分为连续可用性系统和高可用性系统。
- ◆ 连续可用系统基本上与生命安全型系统相对应，因为如果系统的不可用时间超出了系统的响应时间，将导致灾难性后果。
- ◆ 高可用性系统（**High-Availability, HA**）是近年来流行的一类系统，这类系统对可靠性的要求与任务确保型系统相当，失效率在 $10^{-3}$ /小时至 $10^{-6}$ /小时之间。加速器控制系统失效率可达到 $10^{-4}$ /小时

HA系统的失效率和可用度

失效率	可用度	年平均停机时间
$10^{-3}$	99.9%	9小时
$10^{-4}$	99.99%	1小时
$10^{-5}$	99.999%	5分钟
$10^{-6}$	99.9999%	31秒

# 实现可靠性系统的基本方法

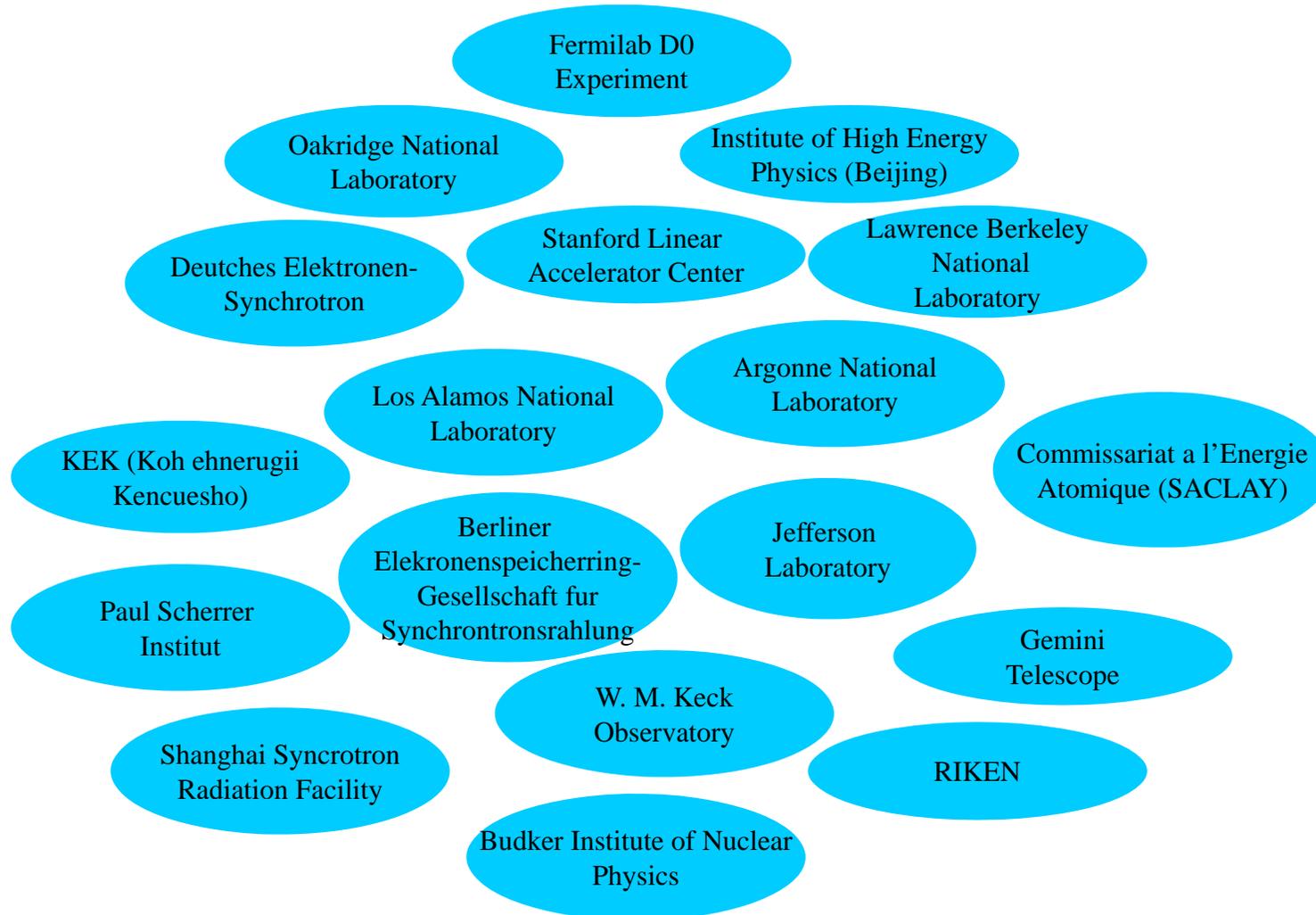
- ◆ 故障：防止或避免系统出现故障（**Fault Prevention or Fault Avoidance**）
  - ❖ 硬件故障的技术包括：选择可靠和经过挑选和处理的元器件、部件和设备；用成熟技术正确地构建系统以及屏蔽系统，防止外界干扰
  - ❖ 系统和软件的防障措施有
    - 仔细定义系统的规格说明，在设计时要对系统进行故障模式及其影响的分析。
    - 采用提供可靠性技术、并经过认证的实时操作系统，将冗余结构的复杂性隐藏在操作系统中。采用可靠性高的软件设计技术
    - 采用良好的开发环境及自动设计工具
- ◆ 容障：使系统在发生故障时仍能正常工作，即容忍故障（**Fault Tolerance**）。
  - ❖ 容障的基本技术是冗余（**redundancy**）。
  - ❖ 空间冗余，又称物理冗余，外加冗余资源减少故障带来的损失；
  - ❖ 时间冗余，通过多执行一遍代码，可以消除暂态故障。此外还有信息冗余，例如通信协议中的冗余码
- ◆ 在设计时，采用验证诊断等技术来排除故障（**Fault Removal**），简称排障。
  - ❖ 排障是在系统实现过程中和实现之后采取的措施。
  - ❖ 核查（**verification**）是在系统从设计到实现的各个阶段中，不断地评价实施方法的正确性。
  - ❖ 验证（**validation**）是在系统完成阶段，采用各种手段来检查其结果的正确性。

# 第三章 分布式控制系统实例

◆ **EPICS ( Experimental Physics and Industrial Control System )** 是免费的大型实验物理和工业控制系统软件包。该软件包最初是由美国洛斯阿拉莫斯国家实验室 (LANL) 和阿贡国家实验室 (ANL) 联合开发的一种控制系统组态软件包，近年来世界上许多实验室加入了这一软件包的开发，因此，**EPICS**系统得到了广泛的应用和飞速发展。本章主要介绍：

- ① 软硬件体系结构
- ② 核心：IOC database + channel access
- ③ 数据库
- ④ 新进展

# Partial List of Collaboration Members



◆ 已被全世界**100**多个大型科学研究工程项目所采用，涉及粒子加速器、大型射电天文望远镜以及飞机工业等领域。

# 硬件体系结构

## Workstations:

Sun  
Hp  
DEC/Alpha  
Silicon Graphics  
PC

## OS:

Unix/Linux Some Windows

## I/O Controllers:

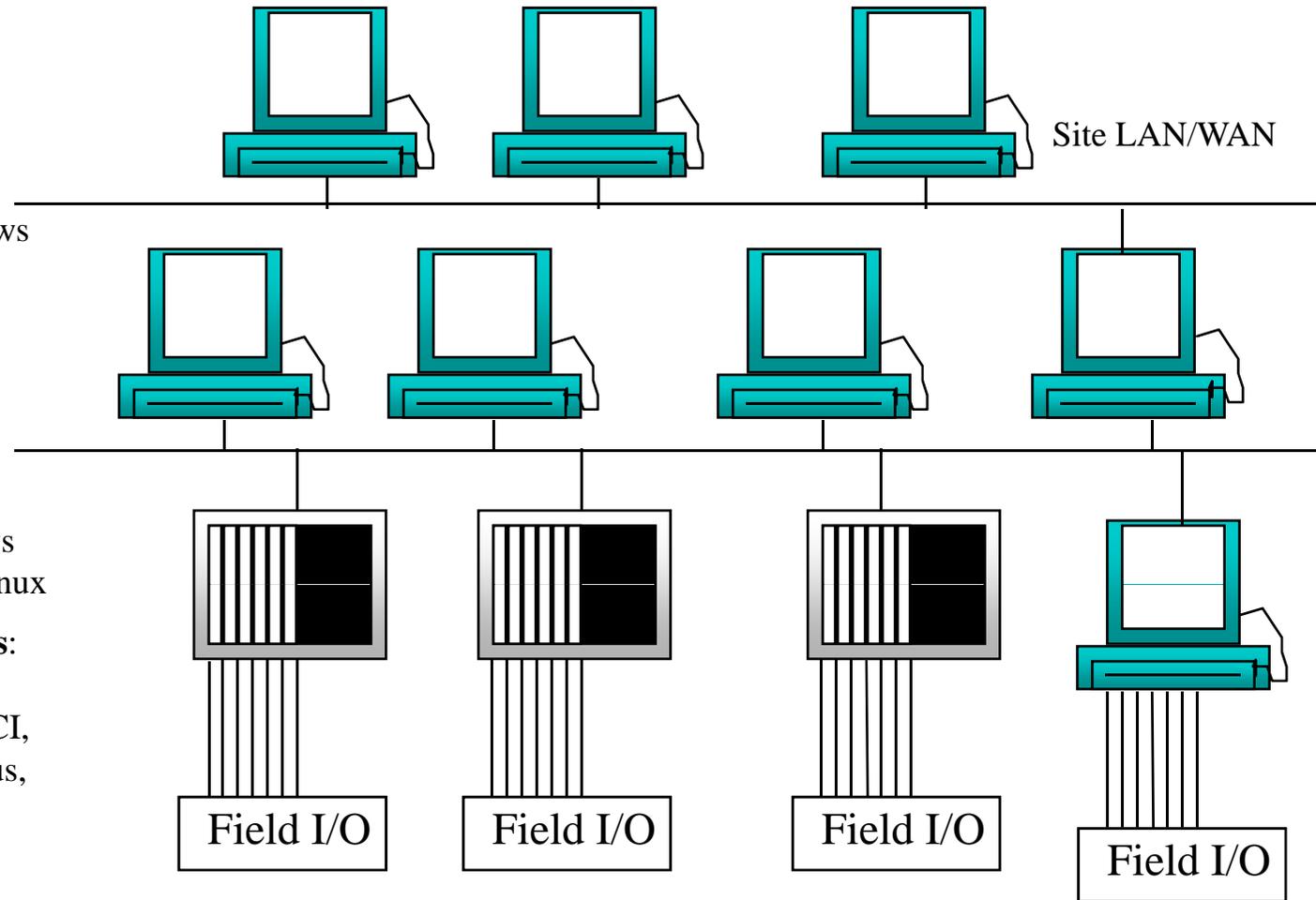
VME, VXI  
PCI, Workstations

## OS:

vxWorks, Unix, Windows  
RTEMS, RTLinux, L4 linux

## Remote and Local I/O Buses:

Control Net, PCI, CAN-Bus,  
Industry Pack, VME, VXI, PCI,  
ISA, CAMAC, GPIB, Profibus,  
Bitbus, Serial, Allen-Bradley,  
Modbus, Yokogawa, G-3,  
Ethernet/IP



# EPICS Architecture

A set of channel access clients for viewing and Modifying data

User Tools  
DM/MEDM/DM2K/EDM/JDM  
Alarm Handler / stripTool/  
knobManager

Connection Server

Archive Viewing | Archive Analysis

Archive Access

Archiving

A data archiver with web based management and a suite of viewing and analysis capabilities

Channel Access Client (CAC)  
Connection | Data Transfers

WAN/LAN/Internet

A plug-in for name resolution to a name server

Connection | Data Transfers  
Channel Access Server (CAS)

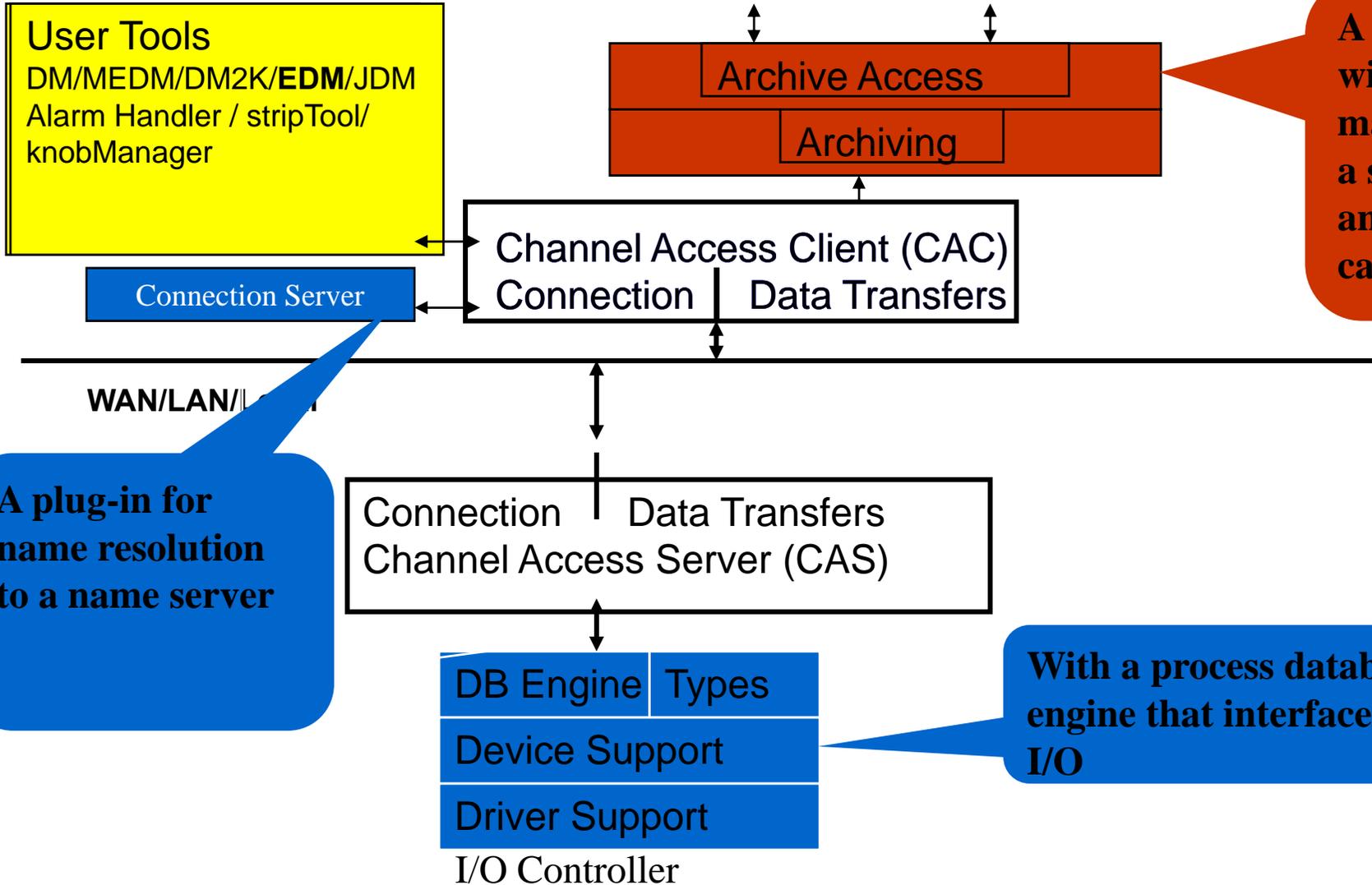
DB Engine | Types

Device Support

Driver Support

I/O Controller

With a process database engine that interfaces to I/O



# EPICS Architecture

A set of channel access clients for viewing and Modifying data

**User Tools**  
 DM/MEDM/DM2K/**EDM**/JDM  
 Alarm Handler / stripTool/  
 knobManager

Archive Viewing | Archive Analysis

Archive Access  
 Archiving

**Program Interfaces**  
 C, C++, Fortran, JAVA  
 IDL / Matlab / Mathematica  
 ActiveX / DDE / VisBasic  
 XAL/ SDDS / SAD /  
 tcl / PERL / Python  
 State Notation Lang / FSQT

Channel Access Client (CAC)  
 Connection | Data Transfers

Connection Server



WAN/LAN/Local

Connection | Data Transfers  
 Channel Access Server (CAS)

Connection | Data Transfers  
 Portable Channel Access Server (CAS)

DB Engine | Types  
 Device Support  
 Driver Support  
 I/O Controller

A plug-in for name resolution to a name server

A portable channel access server that supports the addition of any new data sources into the EPICS environment that is used to integrate many different platforms

**A set of channel access clients for viewing and Modifying data**

# EPICS Architecture

Archive Viewing | Archive Analysis

Archive Access  
Archiving

**User Tools**  
DM/MEDM/DM2K/**EDM**/JDM  
Alarm Handler / stripTool/  
knobManager

**Program Interfaces**  
C, C++, Fortran, JAVA  
IDL / Matlab / Mathematica  
ActiveX / DDE / VisBasic  
XAL/ SDDS / SAD /  
tcl / PERL / Python  
State Notation Lang / FSQT

Connection Server

Channel Access Client (CAC)  
Connection | Data Transfers

WAN/LAN/Local

Connection | Data Transfers  
Channel Access Server (CAS)

Connection | Data Transfers  
Portable Channel Access Server (CAS)

DB Engine | Types  
Device Support  
Driver Support  
I/O Controller

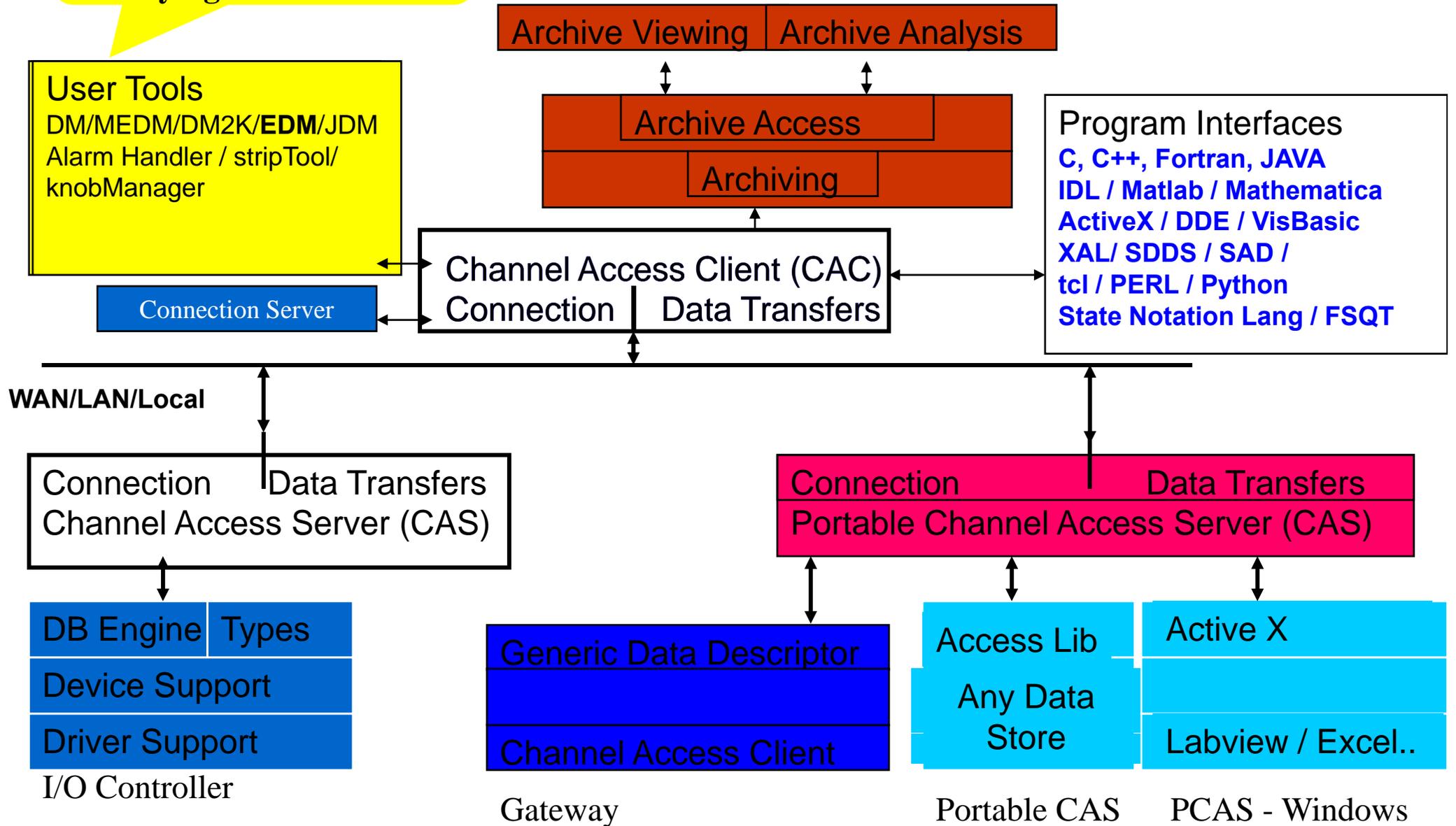
Generic Data Descriptor  
Channel Access Client

Gateway

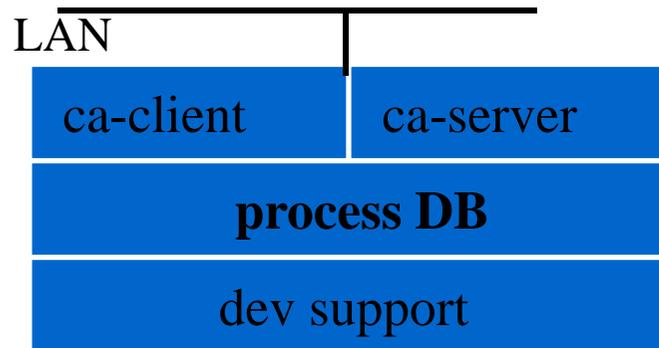
**Including a gateway that isolates network traffic between portions of the control network**

# EPICS Architecture

A set of channel access clients for viewing and Modifying data



# IOC核心



A Channel Access server provides connection, get, put, and monitor services to this database

A Channel Access client provides access to processDBs in other IOCs

Process Blocks are the basic elements: AI, AO, BI, BO, Motor, CALC, PID, SUB, etc....

Process Blocks consist of fields for: SCHEDULE, I/O, CONVERT, ALARM, MONITOR

They hold runtime values: VALUE, TIMESTAMP, ALARM CONDITION, etc....

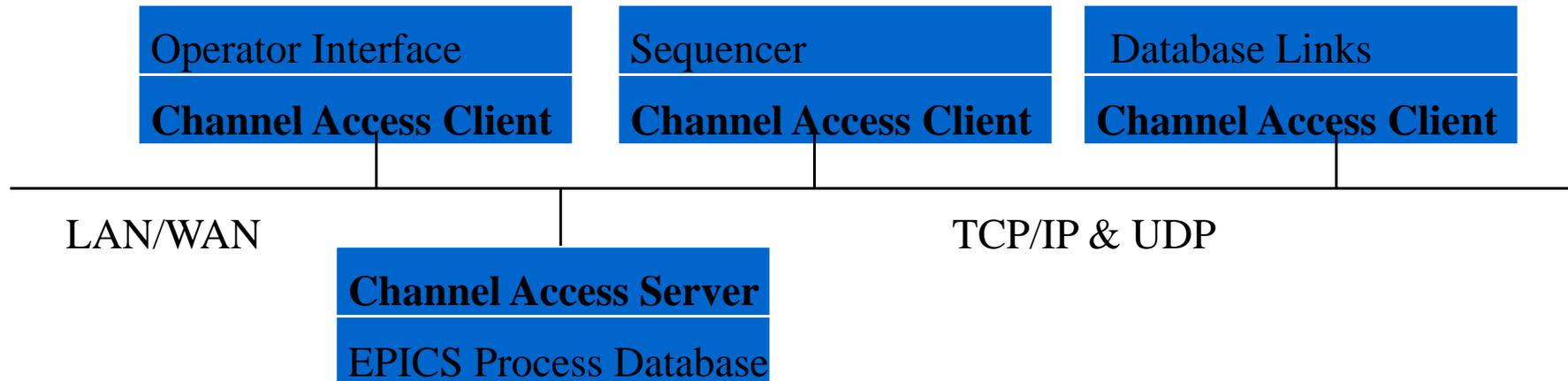
New process block are easily added

Configured using CAPFAST, VDCT, Relational DB, Text Editor at the Host

Loaded as ASCII records into vxWorks at boot time

All fields can be read/written through the channel access client interface during operation

# IOC Core: Channel Access Client/Server Libraries



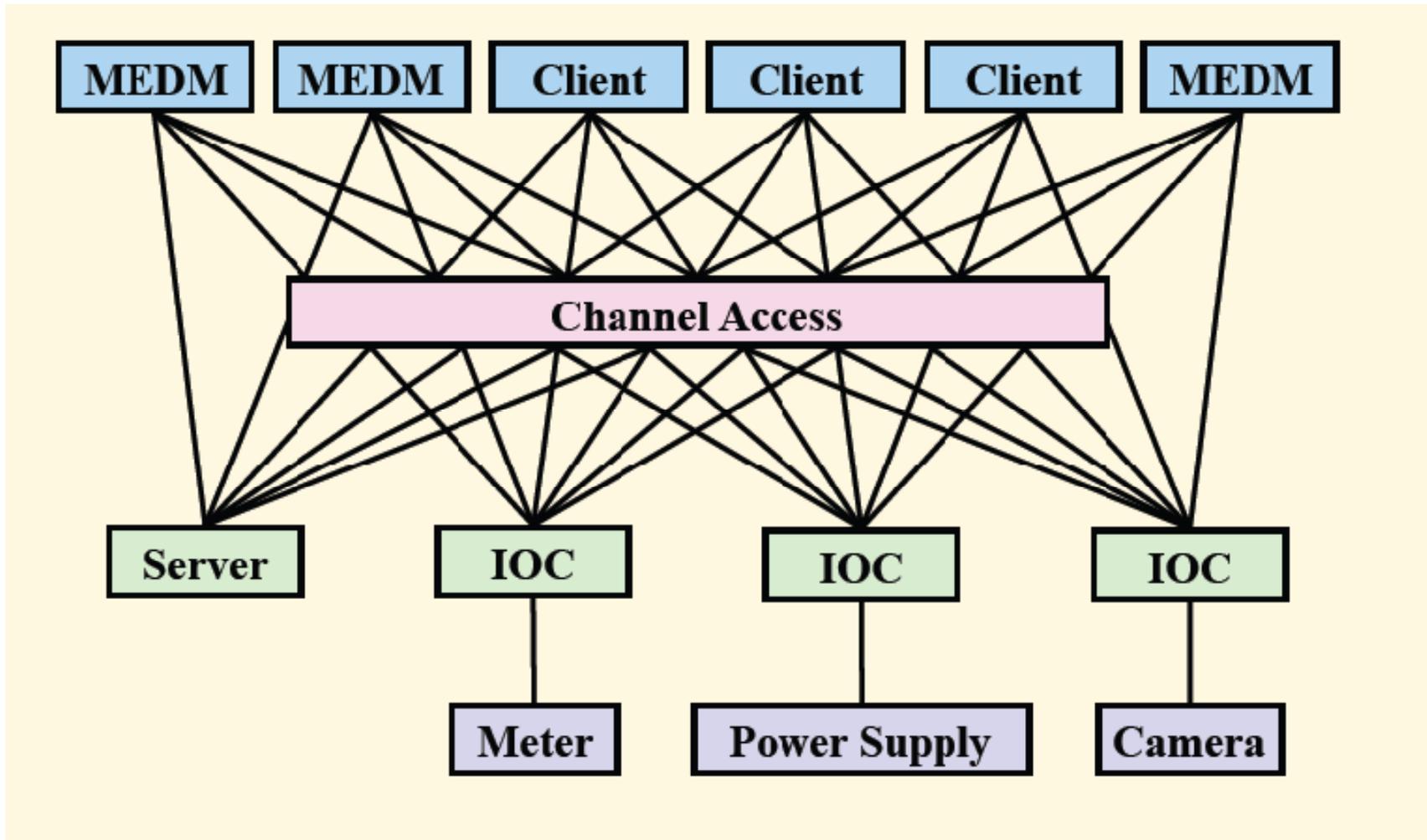
Client: Provides read/write connections to any subsystem on the network with a channel access server

Server: Provides read/write connections to information in this node to any client on the network through channel access client calls. The data resides here!

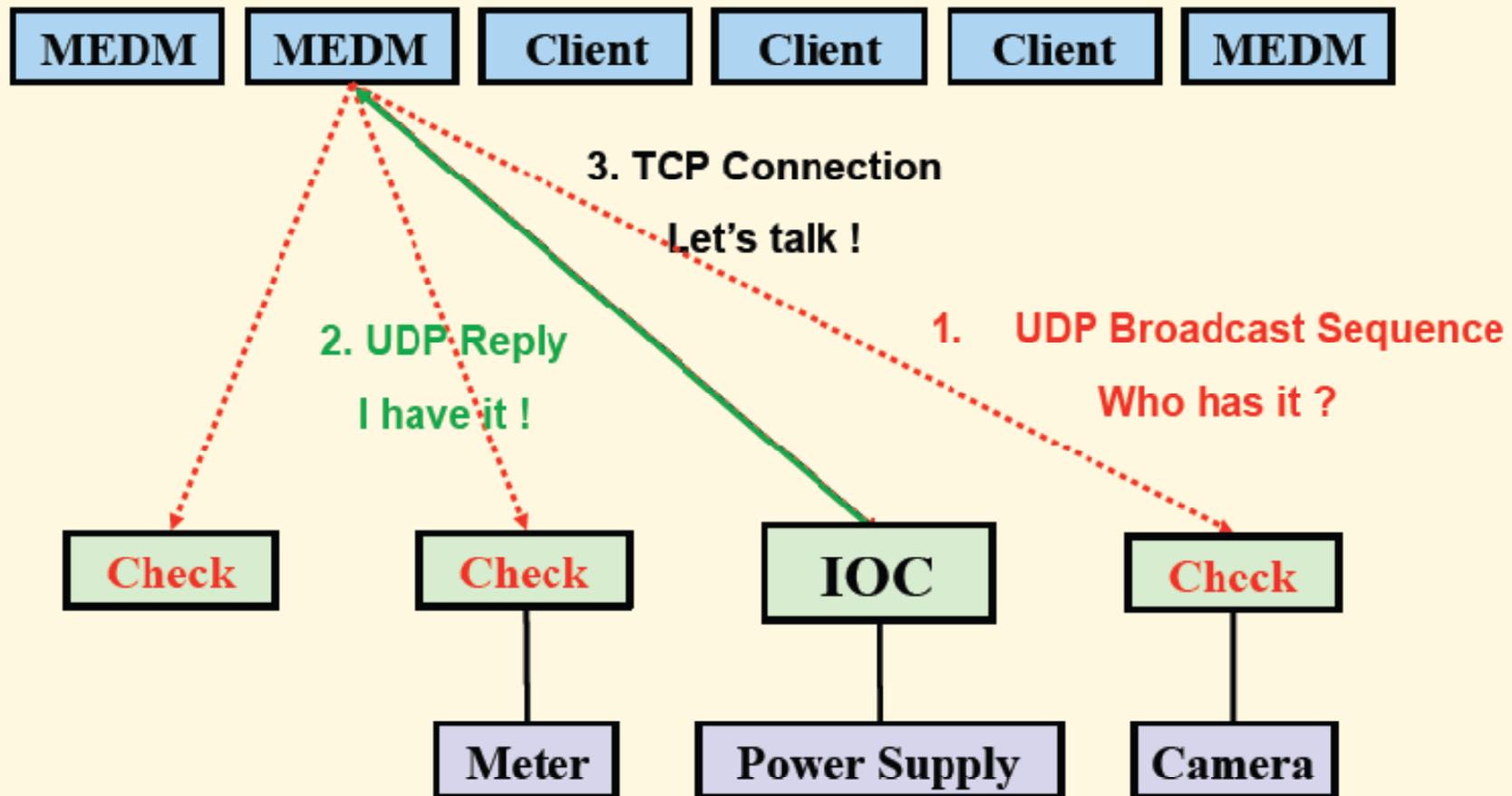
Services: Dynamic Channel Location, Get, Put, Monitor  
Access Control, Connection Monitoring, Automatic Reconnect  
Conversion to client types, Composite Data Structures

Platforms: UNIX/Linux, vxWorks, VMS (Client only), Windows NT, RTEMS, rtLinux

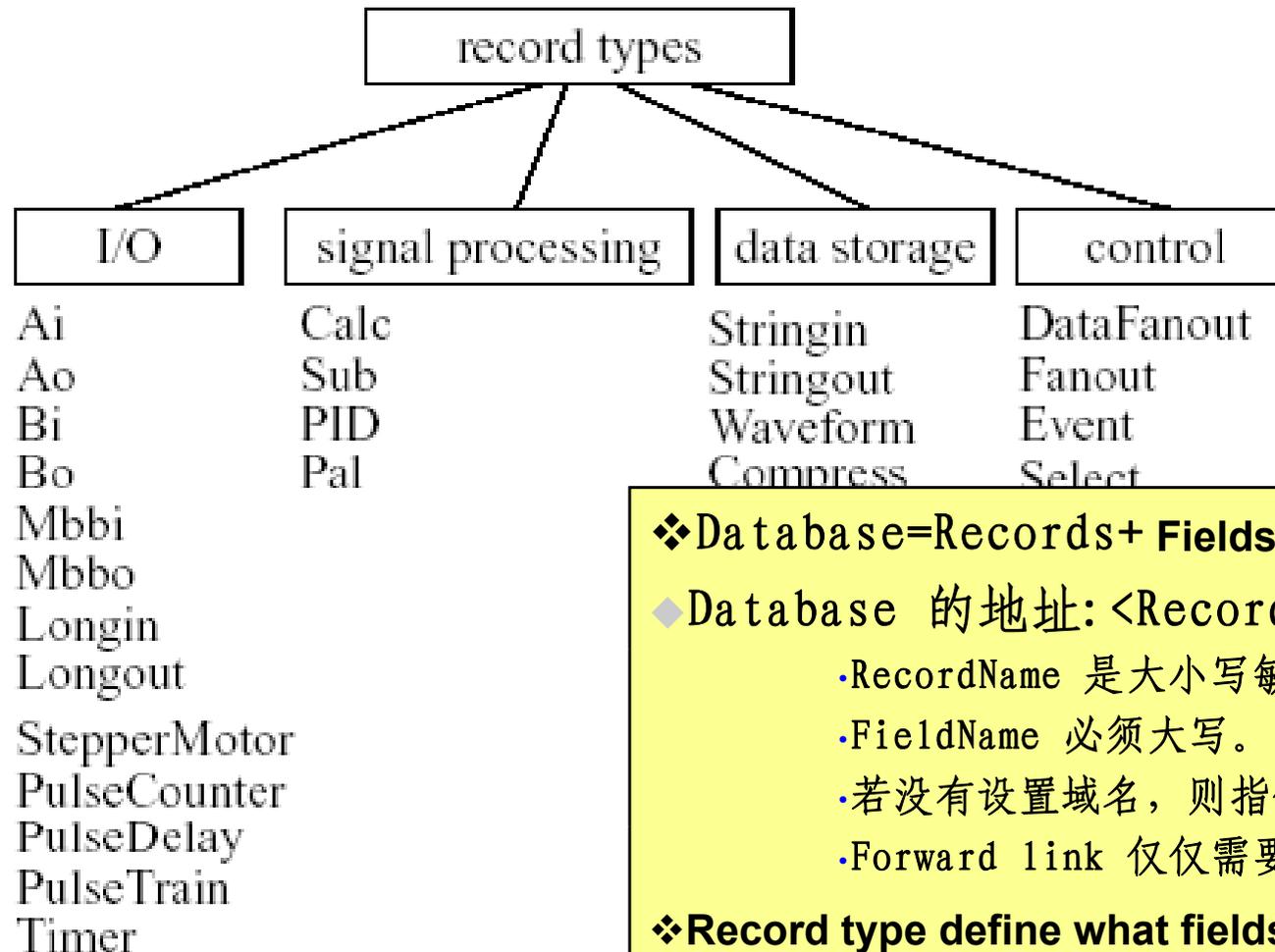
# CA 工作机理



# Search and Connect Procedure



# EPICS数据库



## ❖ Database=Records+ Fields+Links

### ◆ Database 的地址: <RecordName>.<FieldName>

- RecordName 是大小写敏感的。
- FieldName 必须大写。
- 若没有设置域名, 则指值域VAL
- Forward link 仅仅需要设置RecordName

## ❖ Record type define what fields

## ❖ Fields controllable

# EPICS 数据库

---

## ◆ Database Definition File

- ❖ 包含各种类型的记录的定义。

- ❖ Menu

- ❖ Record Type

- ❖ Device

- ❖ Driver

- ❖ Breakpoint Table

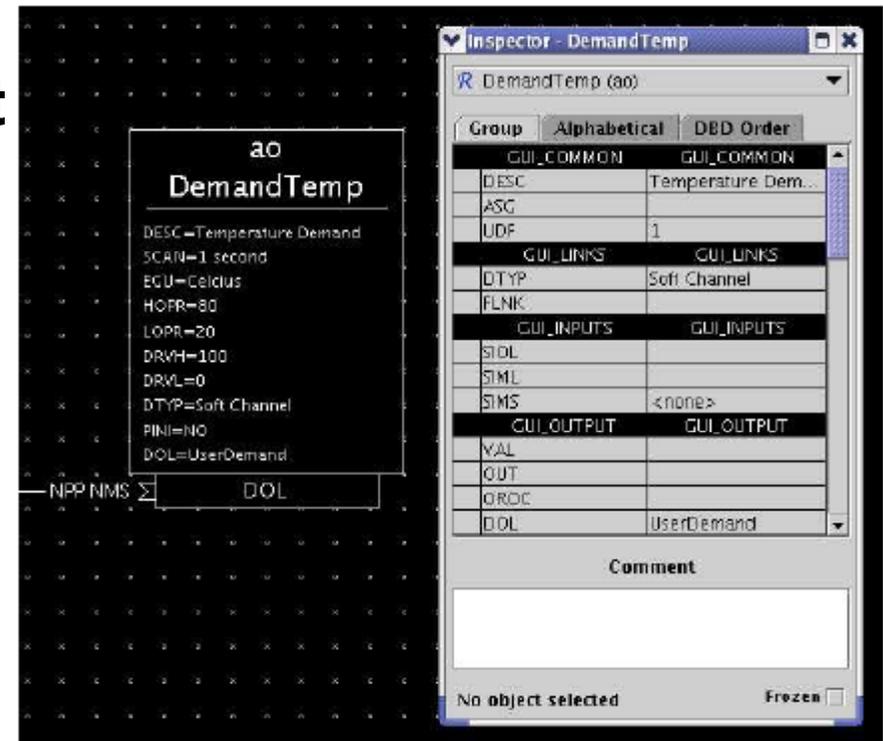
- ❖ Record Instance

## ◆ Record Instance File

- ❖ 仅仅包含记录例子的定义

# Getting started with EPICS

- ◆ Download EPICS 3.14.\* :<http://www.aps.anl.gov/epics/>
- ◆ Install EPICS 3.14
  - ❖ PC/Linux as a host and target
- ◆ Get a EPICS App. Template
- ◆ Configure DB using VDCT
- ◆ Configure IOC startup script st.cmd



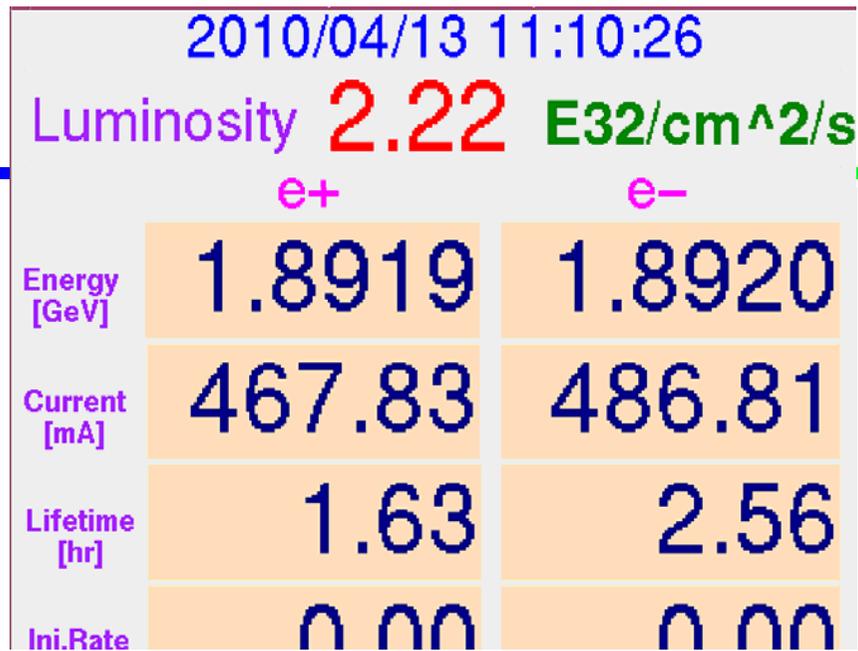
- ◆ Reference:
  - ◆ Record Reference Manual
  - ◆ IOC Application Developer's Guide

# Accelerator Control Systems

Lab	Machine	First User Operation (planned)	First (Low) level systems	Second (Middle) level Systems	Servers	Consoles
PSI	Cyclotron	1974	Camac + Byte Serial Highway	VME + LynxOS	DEC Alpha/VMS	DEC Alpha/VMS
FNAL	Tevatron	1985	Camac		VMS	VMS
RAL	ISIS	1985	Camac + Vsystem		VMS	VMS
CERN	LEP	1988	G64+1553	VME/PPC/LynxOS	HP/HPUX	HP/HPUX
LANL	LANSCE	1988	Camac		VMS	VMS
SLAC	SLC	1989	Camac		VMS	VMS
JLAB	Cebaf	1993	Camac + Serial Highway EPICS	EPICS+VME+VxWorks	HP/HPUX	HP/HPUX
ANL	APS	1997	EPICS+VME/68K +VxWorks+100baseT		Sun	Sun
BESSY	BessyII	1998	SBC+CAN EPICS	EPICS+VME+VWorks	HP/HPUX	HP/HPUX
KEK	KEK-B	1999	EPICS+VXI/VME		HP	HP
BNL	RHIC	2000	VME(~185crtaes)+VxWorks+RPC	Cdev	Sun+Linux	Sun+Linux
PSI	SLS	2000	EPICS+VME64X+ PPC+VxWorks+100baseT	N/A	PC/Linux(10)	PC/Linux
SLAC	SPEARIII	(2003)	EPICS+VME+Rtems			
University of Saskatchewan	CLS	(2003)	EPICS+SBC+Rtems PC104+Linux+EPICS VME I/O+PC cntroller	PC	PC/Linux	PC/Linux
ORNL	SNS	(2006)	EPICS+VME64X/VXI +PPC2100+VxWorks EPICS+PC+Windows AB ControlLogix PLCs+Beckhoff/FlexIO /G3/Ethernet(Fieldbuses)	NA	PC Linux	PC Linux
JAEA	J-PARC	(2007)	EPICS+VME +PPC+VxWorks EPICS+PC/Linux PLCs+/Ethernet(Fieldbuses)	NA	PC Linux	PC Linux

# New Developments

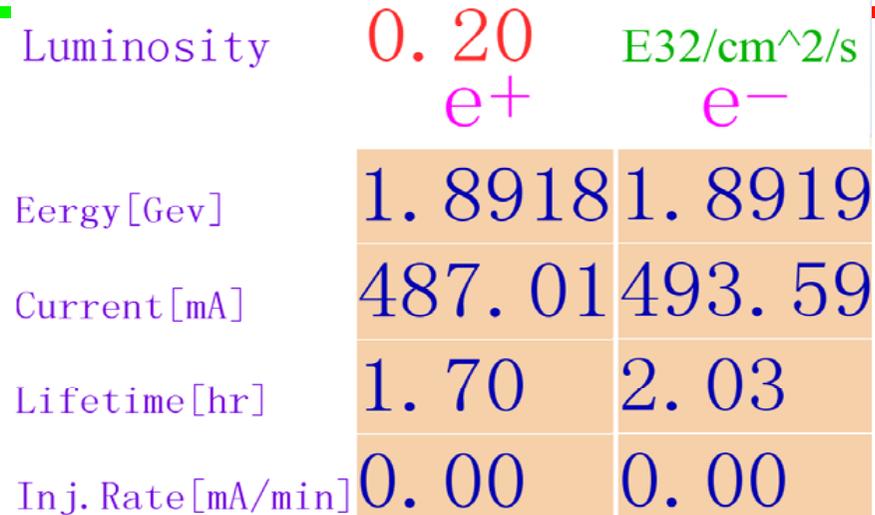
- ◆ 随着互联网和计算机技术的快速发展，通过**Web**浏览器对远程数据和画面进行监控受到越来越广泛的应用。近年来，在加速器研究领域，国外各大实验室正致力于将**Web**技术引入加速器广泛应用的**EPICS**控制系统中，基于**Web**的远程监控系统将成为对加速器运行情况监控的主流。
- ◆ **WebCA/CAML**
  - ❖ 是一种基于**B/S**模式的**EPICS**监控界面实现软件
  - ❖ 由美国**SNS**的**Tom Pelaia**研制
  - ❖ 第一阶段是**Web CA**插件的建立，**Web CA**插件为**EPICS** 通道访问客户端调用提供**JAVA**脚本的应用程序接口。
  - ❖ 第二阶段引入**CAML**，用于在网页中布局控制器。
  - ❖ 现在，**CAML**正处于第三发展阶段，主要是将其已实现和不稳定的**功能**进行进一步的完善和改进，同时再增加一些新功能。
- ◆ 目前，**SNS**和**SLAC**正在推广和使用**CAML**



EDM实现的监控界面

# BEPCII参数监控界面

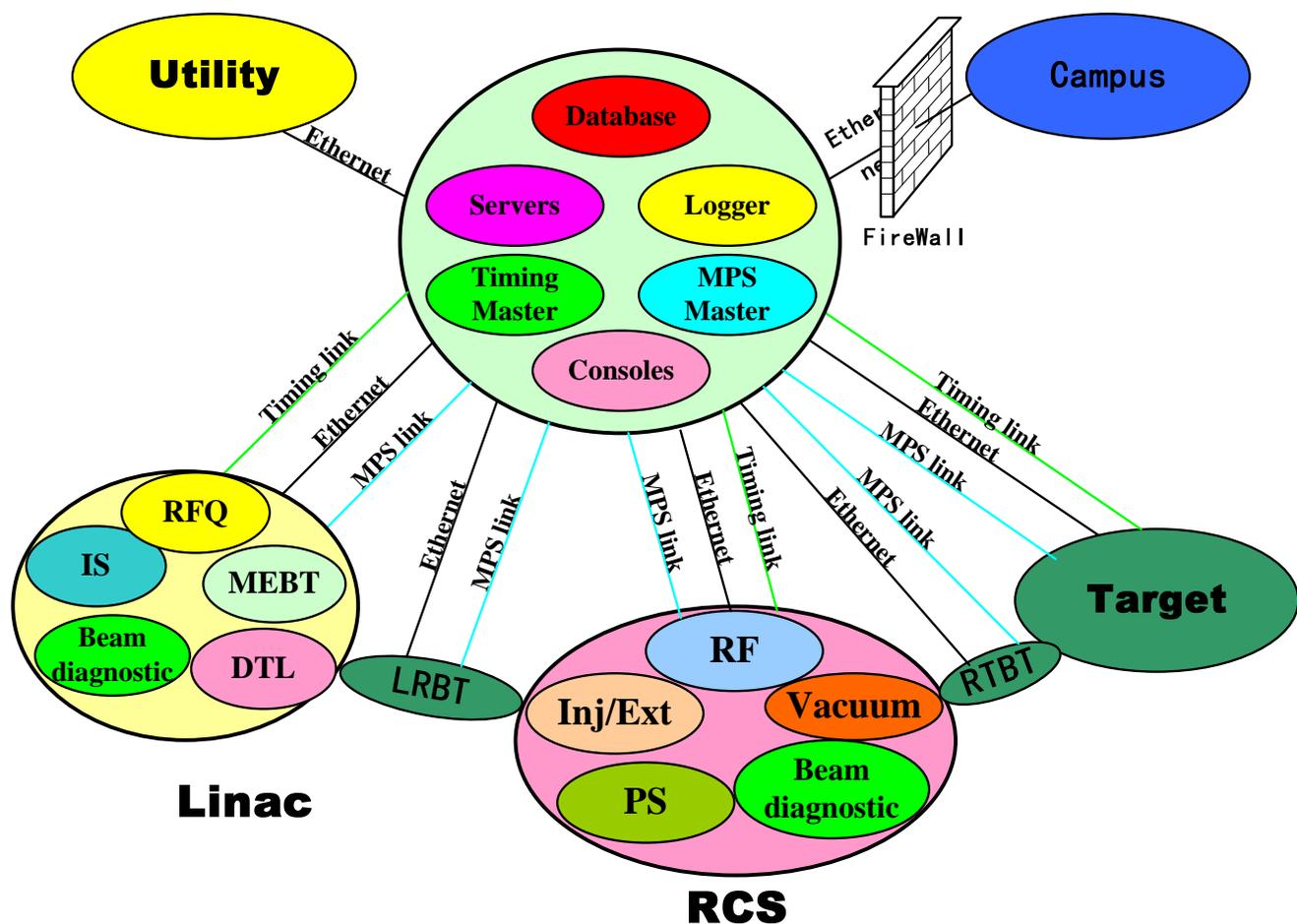
2010-4-20 10:52:58 PM



CAML实现的监控界面



# 第四章 CSNS控制系统设计

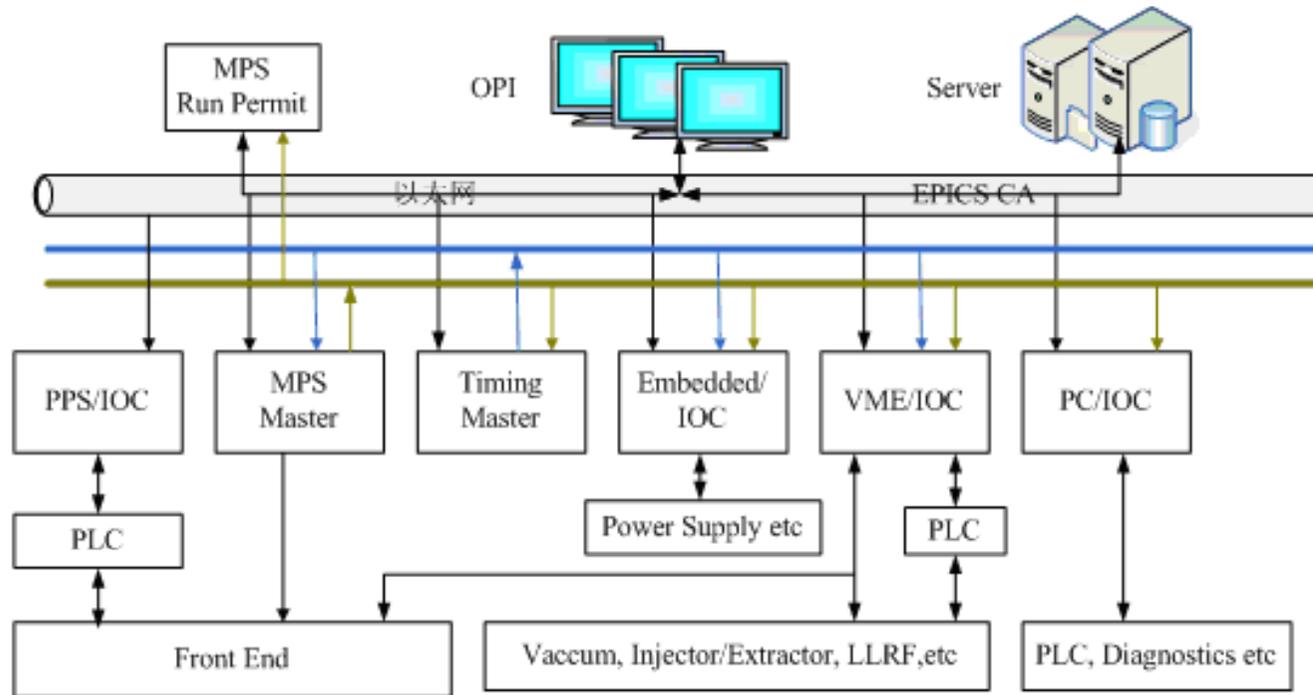


- ◆ 强流质子加速器
- ◆ 控制束流损失和放射性
- ◆ console, computers/servers
- ◆ network, database,
- ◆ physics application software
- ◆ devices control
- ◆ MPS and timing
- ◆ control or data acquisition for the target and the experimental stations will be done using EPICS by the target people.
- ◆ PPS is a separate system that will be monitored by the control system.

# Controlled Devices and Points

item	Device	Qt.	AI	AO	DI	DO	WF	Other	Sum
1	Power Supply	223	244	244	1195	956	7	/	2885
2	Vacuum	190	400	200	430	550	380	30	2180
3	Injection Kicker	2	8	8	20	10	2	/	50
4	Extraction Kicker	7	14	14	20	10	14	/	79
5	RCS LLRF	9	27	27	50	50	9	/	172
6	Beam Diagnostic	300	/	/	/	/	/	/	300
7	Ion Source	10	30	10	28	12	/		80
8	Linac LLRF	6	6	4	16	6	/	/	32
Sum		747	715	505	1805	2071	412		5478

# System Architecture



Two layer architecture

1. OPI:

PC/Linux/Windows

2. IOC

VME IOC

PC/IOC

Embedded IOC

no field bus to a third layer

extensive use:

a field bus or serial interfaces from the lower layer to the equipments.

- ◆ Early IOCs :MVME5100(PPC604) applied in the BEPCII control system.
- ◆ The feasibility study of MVME5500/MVME6100 running RTEMS has started since September of 2005.
- ◆ IOCs will be installed for some technical systems.
- ◆ The preferred interfaces: analogue,digital and serial (RS232, RS485 etc).

# System Architecture

---

- ◆ **A distributed system based on Ethernet**
- ◆ **Adopt the standard two-layer architecture**
  - ❖ **PC/Linux/Windows as the client**
  - ❖ **VME IOC/PC IOC/Embedded IOC as the server.**
- ◆ **EPICS development environment.**
  - ❖ **EPICS 3.14.9**
    - **VxWorks 5.4.2 (used in the BEPCII, in the early development)**
    - **VxWorks 6.5 (expensive, under consideration)**
    - **RTEMS (free, open-source, under study, need powerful experts)**

# Feasibility Study

---

- ◆ **The control system conceptual design was reviewed in the CSNS international review meeting in May of 2007.**
- ◆ **Since May of 2007, the control system has started R&D prototype with very limited budget.**
- ◆ **There were totally 5 prototypes:**
  - ❖ **A WE7000 for the injection PS control**
  - ❖ **An Ion source control using YOKOGAWA PLC FA-M3**
  - ❖ **A fast waveform monitor and verifier for a dipole PS**
  - ❖ **Embedded ColdFire IOC for DPSCM**
  - ❖ **EVG/EVR Timing.**

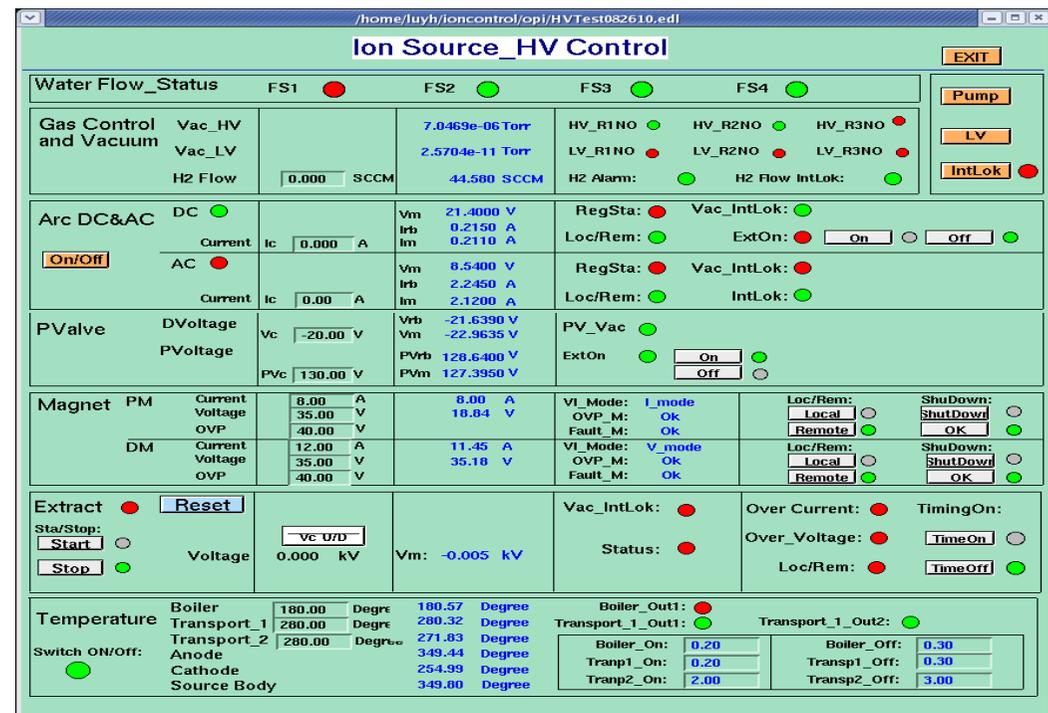
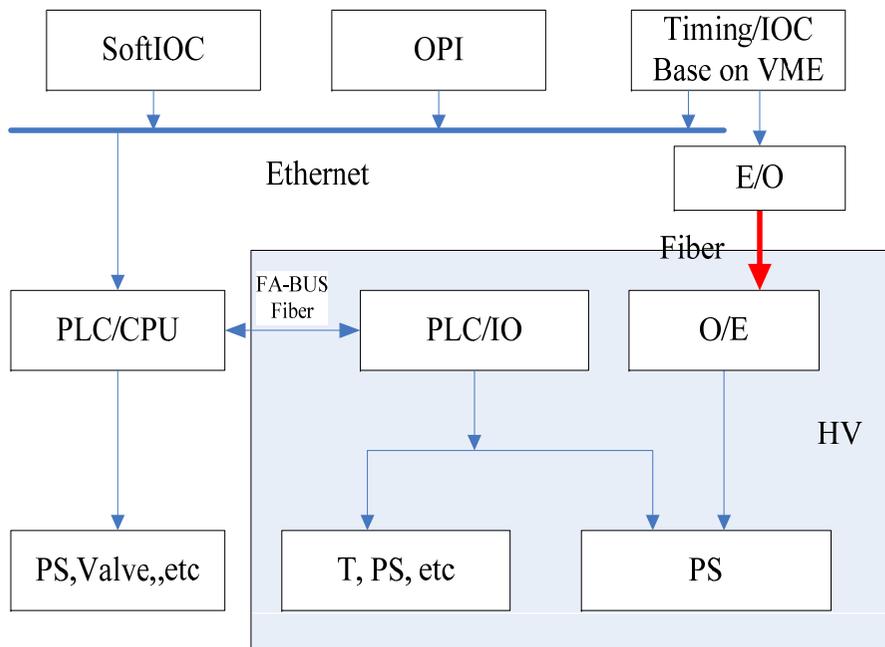
# Front-end Control Interface

---

- ◆ **Front-end includes an ion source and LEBT. The ion source control system is typically a single component of the CSNS control system.**
- ◆ **It includes 5 parts:**
  - ❖ **power supplies, vacuum, temperature measurement, water cooling and timing.**
- ◆ **All are required to be controlled locally in the tunnel and remotely at the central control room.**
- ◆ **The power supplies work in the high voltage environment.**
- ◆ **Require the control system with high reliability and availability.**

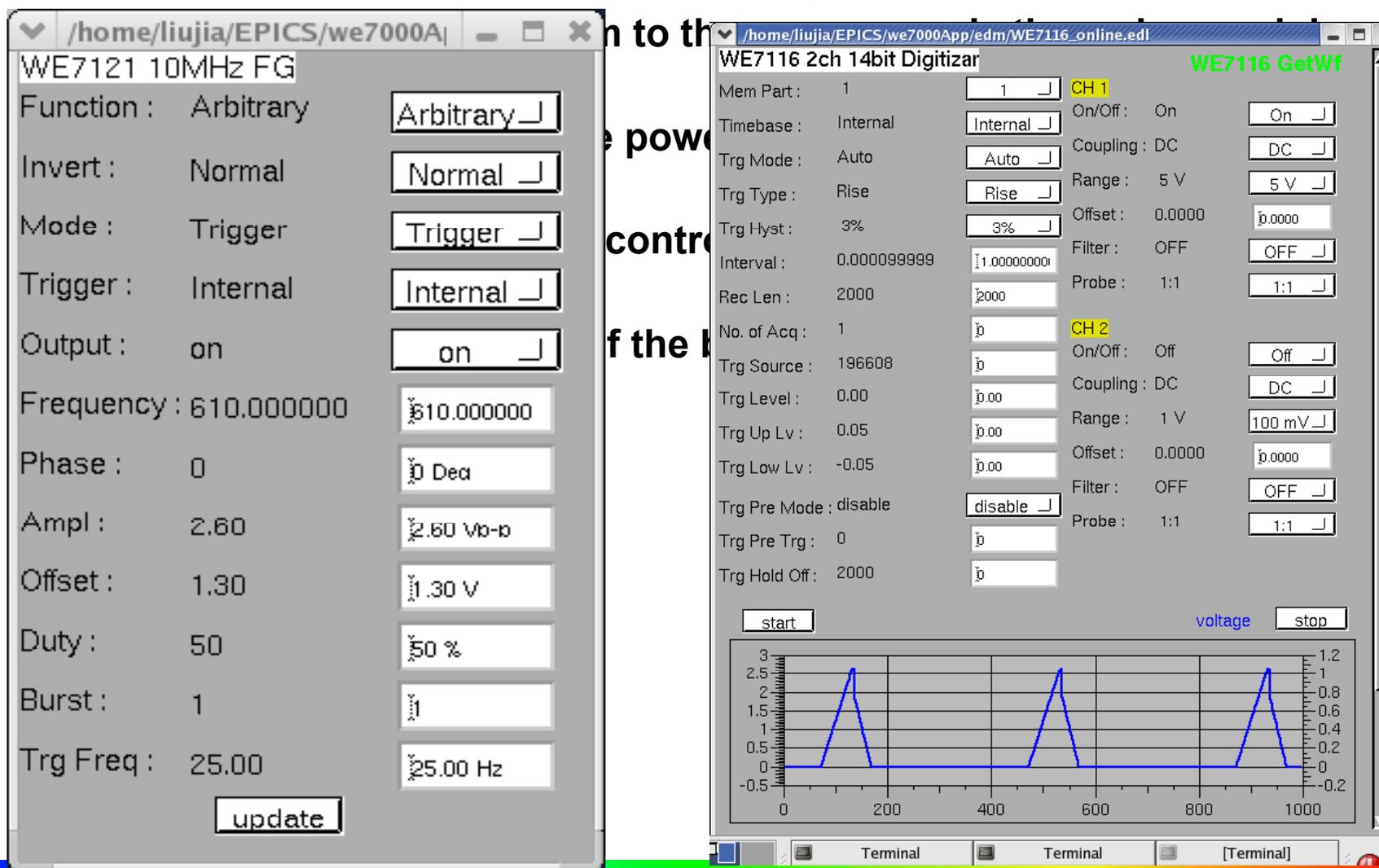
# Prototype of Ion Source Control

- ◆ A PC/Linux running softIOC exchanges the data with PLC/CPU via Ethernet.
- ◆ The PLC/CPU communicates with the PLC/IO (all inputs and outputs) modules through the FA-Bus Optical Fiber.
- ◆ No extra electrical isolation between the ground and high voltage platform.



# Injection Control

- ◆ The YOGOKAWA WE7000 measurement system is used



# High Application Software

---

- ◆ **Adopt XAL:** has been used for SNS commissioning and operation for over four years.
- ◆ **Some XAL work in CSNS has been done**
  - ❖ Established XAL for the CSNS accelerator
  - ❖ Database framework of the CSNS accelerator by using standardized rules and interfaces
  - ❖ Imported most of the equipments data of the CSNS Linac accelerator and two beam transfer lines
  - ❖ A virtual accelerator has been successfully setup.
- ◆ **Provide a data platform for testing the XAL accelerator physics application before the construction of the accelerator.**
- ◆ **In the future, it can be easily expanded to include all the CSNS accelerator sections.**

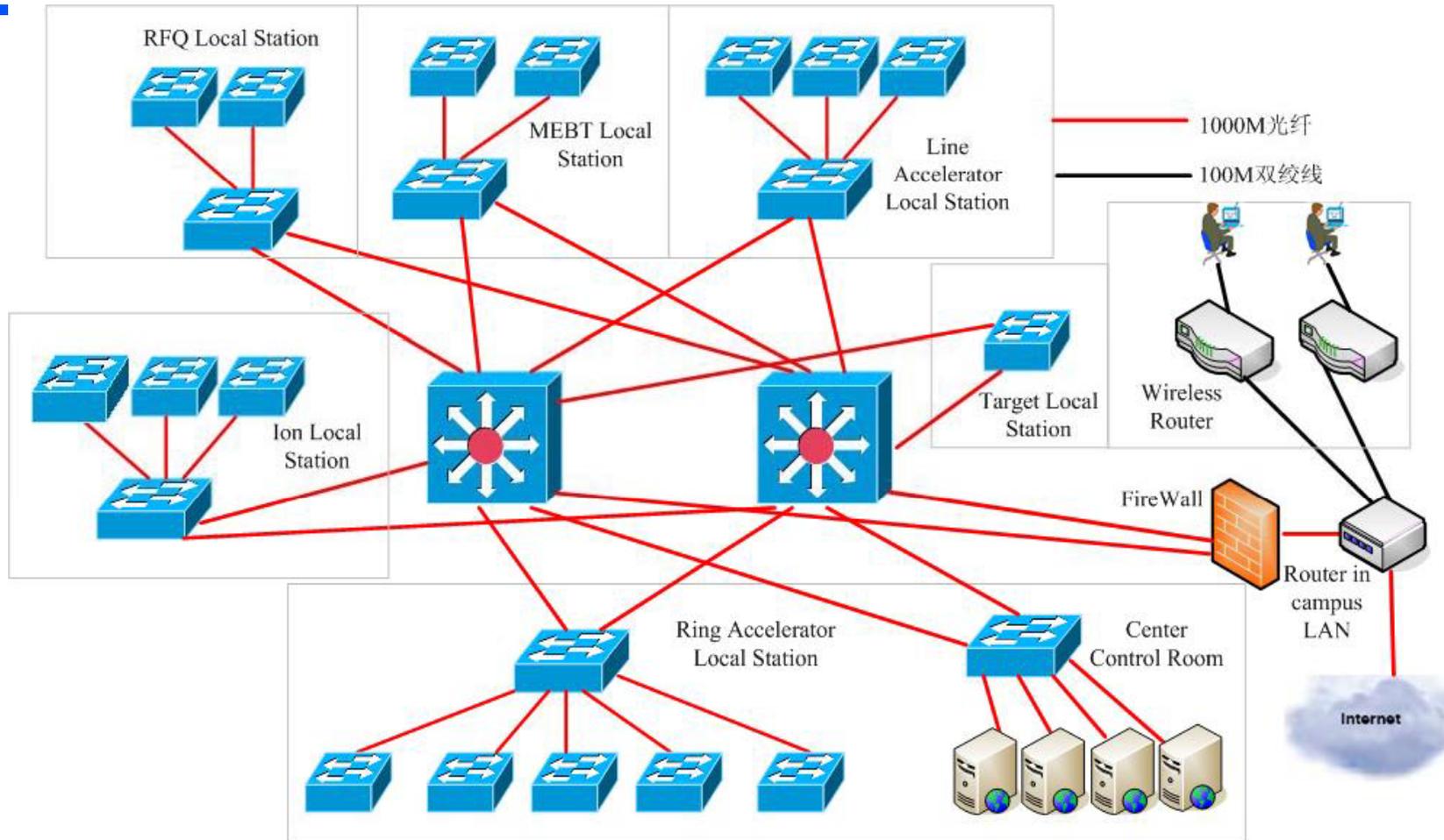
# CSNS XAL

The image shows two overlapping software windows. The background window is 'DB2XAL - Untitled.db2xal' with a menu bar (File, Edit, View, Window, Help) and buttons for 'Connect', 'Refresh Data', and 'Beamline ID: 2.0'. It features a 'Select Sequences' table and a 'Database Status' message at the bottom.

The foreground window is 'Virtual Accelerator - (MEBT) - Untitled.va\*' with a menu bar (File, Edit, Accelerator, Mode, View, Window, Help) and buttons for 'Probe File...', 'Probe Editor', 'Set Noise...', 'Start VA', and 'Stop VA'. It displays a data table with columns for Node, Readback PV, Readback, Setpoint PV, and Setpoint.

Node	Readback PV	Readback	Setpoint PV	Setpoint
MEBT_MAG:QH01	MEBT_MAG:QH01:B	32.0	MEBT_MAG:PS_QH01:B_Set	32.0
MEBT_MAG:QV01	MEBT_MAG:QV01:B	27.0	MEBT_MAG:PS_QV01:B_Set	27.0
MEBT_MAG:QH02	MEBT_MAG:QH02:B	17.5	MEBT_MAG:PS_QH02:B_Set	17.5
MEBT_MAG:QV02	MEBT_MAG:QV02:B	12.0	MEBT_MAG:PS_QV02:B_Set	12.0
MEBT_MAG:QH03	MEBT_MAG:QH03:B	13.820074	MEBT_MAG:PS_QH03:B_Set	13.820074
MEBT_MAG:QV03	MEBT_MAG:QV03:B	20.172169	MEBT_MAG:PS_QV03:B_Set	20.172169
MEBT_MAG:QH04	MEBT_MAG:QH04:B	24.634729	MEBT_MAG:PS_QH04:B_Set	24.634729
MEBT_MAG:QV04	MEBT_MAG:QV04:B	16.504729	MEBT_MAG:PS_QV04:B_Set	16.504729
DTL1	DTL1:cavAmpAvg	12.3964	DTL1:AmpSet	12.3964
DTL1	DTL1:cavPhaseAvg	-30.09	DTL1:PhaseSet	-30.09
DTL_MAG:QV001	DTL_MAG:QV001:B	7.5	DTL_MAG:PS_QV001:B_Set	7.5
DTL_MAG:QH001	DTL_MAG:QH001:B	7.5	DTL_MAG:PS_QH001:B_Set	7.5
DTL_MAG:QV002	DTL_MAG:QV002:B	7.5	DTL_MAG:PS_QV002:B_Set	7.5
DTL_MAG:QH002	DTL_MAG:QH002:B	7.5	DTL_MAG:PS_QH002:B_Set	7.5
DTL_MAG:QV003	DTL_MAG:QV003:B	7.5	DTL_MAG:PS_QV003:B_Set	7.5
DTL_MAG:QH003	DTL_MAG:QH003:B	7.5	DTL_MAG:PS_QH003:B_Set	7.5
DTL_MAG:QV004	DTL_MAG:QV004:B	7.5	DTL_MAG:PS_QV004:B_Set	7.5
DTL_MAG:QH004	DTL_MAG:QH004:B	7.5	DTL_MAG:PS_QH004:B_Set	7.5
DTL_MAG:QV005	DTL_MAG:QV005:B	7.5	DTL_MAG:PS_QV005:B_Set	7.5
DTL_MAG:QH005	DTL_MAG:QH005:B	7.5	DTL_MAG:PS_QH005:B_Set	7.5
DTL_MAG:QV006	DTL_MAG:QV006:B	7.5	DTL_MAG:PS_QV006:B_Set	7.5
DTL_MAG:QH006	DTL_MAG:QH006:B	7.5	DTL_MAG:PS_QH006:B_Set	7.5
DTL_MAG:QV007	DTL_MAG:QV007:B	7.5	DTL_MAG:PS_QV007:B_Set	7.5
DTL_MAG:QH007	DTL_MAG:QH007:B	7.5	DTL_MAG:PS_QH007:B_Set	7.5
DTL_MAG:QV008	DTL_MAG:QV008:B	7.5	DTL_MAG:PS_QV008:B_Set	7.5
DTL_MAG:QH008	DTL_MAG:QH008:B	7.5	DTL_MAG:PS_QH008:B_Set	7.5
DTL_MAG:QV009	DTL_MAG:QV009:B	7.5	DTL_MAG:PS_QV009:B_Set	7.5
DTL_MAG:QH009	DTL_MAG:QH009:B	7.5	DTL_MAG:PS_QH009:B_Set	7.5
DTL_MAG:QV010	DTL_MAG:QV010:B	7.5	DTL_MAG:PS_QV010:B_Set	7.5
DTL_MAG:QH010	DTL_MAG:QH010:B	7.5	DTL_MAG:PS_QH010:B_Set	7.5
DTL_MAG:QV011	DTL_MAG:QV011:B	7.5	DTL_MAG:PS_QV011:B_Set	7.5
DTL_MAG:QH011	DTL_MAG:QH011:B	7.5	DTL_MAG:PS_QH011:B_Set	7.5
DTL_MAG:QV012	DTL_MAG:QV012:B	7.5	DTL_MAG:PS_QV012:B_Set	7.5
DTL_MAG:QH012	DTL_MAG:QH012:B	7.5	DTL_MAG:PS_QH012:B_Set	7.5
DTL_MAG:QV013	DTL_MAG:QV013:B	7.5	DTL_MAG:PS_QV013:B_Set	7.5
DTL_MAG:QH013	DTL_MAG:QH013:B	7.5	DTL_MAG:PS_QH013:B_Set	7.5
DTL_MAG:QV014	DTL_MAG:QV014:B	7.5	DTL_MAG:PS_QV014:B_Set	7.5
DTL_MAG:QH014	DTL_MAG:QH014:B	7.5	DTL_MAG:PS_QH014:B_Set	7.5
DTL_MAG:QV015	DTL_MAG:QV015:B	7.5	DTL_MAG:PS_QV015:B_Set	7.5
DTL_MAG:QH015	DTL_MAG:QH015:B	7.5	DTL_MAG:PS_QH015:B_Set	7.5
DTL_MAG:QV016	DTL_MAG:QV016:B	7.5	DTL_MAG:PS_QV016:B_Set	7.5

# Topology of the Control System Network



- ◆ 100Mbit switched Ethernet with a Gigabit switched Ethernet backbone.
- ◆ EPICS CA gateway for the different IOC PV access and effective management of traffic and security.

# Reference

- ◆ 《自动控制原理》，胡寿松主编，科学出版社
- ◆ 《计算机控制系统的原理和方法》，刘松强编著，科学出版社
- ◆ 《实时计算机系统》，刘松强编著，学苑出版社
- ◆ 《粒子加速器技术》，赵籍九，尹兆升主编，高等教育出版社
- ◆ “高速串行互连技术在VME/CPCI总线中的应用及发展”，赵基海，大地信合（北京）科技有限公司，2007
- ◆ EPICS web site: <http://www.aps.anl.gov/epics/>
- ◆ Bob Dalesio. Experimental Physics and Industrial Control System (EPICS) Overview. Jun. 2001
- ◆ Kazuro Furukawa, “Introduction to Channel Access Client Library”, EPICS training course, IHEP, Beijing, March, 2010
- ◆ Andrew Johnson. EPICS Database Principles. Jan. 2007
- ◆ Matej Sekoranja. VisualDCT User’s Manual. Cosylab Ltd., Jul. 2003
- ◆ . F. Lambiase, B. Oerter, J. Smith. Power Supply Control for the Spallation Neutron Source. EPAC’00, Vienna, Austria, June 2000.
- ◆ “VMEbus Handbook”, Wade Peterson, published by VITA
- ◆ VMEbus International Trade Association (VITA) web site: <http://www.vita.com>
- ◆ 《VxWorks BSP开发人员指南》Wind River著，王金刚，苏琪，杨锡劭. . 清华大学出版社，2003
- ◆ 罗修波. 实时操作系统VxWorks的内核任务调度研究. 电子开发网，2008
- ◆ <http://www.ornl.gov/~t6p/Main/Welcome.html>
- ◆ J. Galambos, C. Chu, S. Cousineau, V. Danilov, J. Patton, T. Pelaia, A. Shishlo, “XAL APPLICATION PROGRAMMING STRUCTURE”, Particle Accelerator Conference, 2005
- ◆ <http://sns.gov/APGroup/appProg/xal/xal.htm>, “XAL description”