冯·诺依曼体系结构冯·诺依曼理论的要点是：数字计算机的数制采用二进制；计算机应该按照程序顺序执行。

根据冯·诺依曼体系结构构成的计算机，必须具有如下功能：把需要的程序和数据送至计算机中。必须具有长期记忆程序、数据、中间结果及最终运算结果的能力。能够完成各种算术、逻辑运算和数据传送等数据加工处理的能力。能够根据需要控制程序走向，并能根据指令控制机器的各部件协调操作。能够按照要求将处理结果输出给用户。

为了完成上述的功能，计算机必须具备五大基本组成部件，包括：输入数据和程序的输入设备、记忆程序和数据的存储器、完成数据加工处理的运算器、控制程序执行的控制器、输出处理结果的输出设备。

冯·诺依曼体系结构

(1)采用存储程序方式，指令和数据不加区别混合存储在同一个存储器中，（数据和程序在内存中是没有区别的,它们都是内存中的数据,当EIP指针指向哪 CPU就加载那段内存中的数据,如果是不正确的指令格式,CPU就会发生错误中断. 在现在CPU的保护模式中,每个内存段都其描述符,这个描述符记录着这个内存段的访问权限(可读,可写,可执行).这最就变相的指定了哪个些内存中存储的是指令哪些是数据）

指令和数据都可以送到运算器进行运算，即由指令组成的程序是可以修改的。

(2)存储器是按地址访问的线性编址的一维结构，每个单元的位数是固定的。

(3)指令由操作码和地址组成。操作码指明本指令的操作类型,地址码指明操作数和地址。操作数本身无数据类型的标志，它的数据类型由操作码确定。

(4)通过执行指令直接发出控制信号控制计算机的操作。指令在存储器中按其执行顺序存放，由指令计数器指明要执行的指令所在的单元地址。指令计数器只有一个，一般按顺序递增，但执行顺序可按运算结果或当时的外界条件而改变。

(5)以运算器为中心，I/O设备与存储器间的数据传送都要经过运算器。

(6)数据以二进制表示。

从本质上讲，冯.诺依曼体系结构的本征属性就是二个一维性，即一维的计算模型和一维的存储模型，简单地说“存储程序”是不确切的。而正是这二个一维性，成就了现代计算机的辉煌，也限制了计算机的进一步的发展，真可谓“成也冯，败也冯”。

冯·诺依曼计算机的软件和硬件完全分离，适用于作数值计算。这种计算机的机器语言同高级语言在语义上存在很大的间隔，称之为冯.依曼语义间隔。造成这个差距的其中一个重要原因就是存储器组织方式不同，冯·诺依曼机存储器是一维的线性排列的单元，按顺序排列的地址访问。而高级语言表示的存储器则是一组有名字的变量，按名字调用变量，不考虑访问方法，而且数据结构经常是多维的（如数组，表格）。另外，在大多数高级语言中，数据和指令截然不同，并无指令可以像数据一样进行运算操作的概念。同时，高级语言中的每种操作对于任何数据类型都是通用的，数据类型属于数据本身，而冯.诺依曼机的数据本身没有属性标志，同一种操作要用不同的操作码来对数据加以区分。这些因素导致了语义的差距。如何消除如此大的语义间隔，这成了计算机面临的一大难题和发展障碍。

为什么现在的计算机转化成以储存器为中心？在微处理器问世之前，运算器和控制器是两个分离的功能部件，加上当时的存储器还是以磁芯存储器为主，计算机存储的信息量较少，因此早期冯·诺依曼提出的计算机结构是以运算器为中心的，其他部件通过运算器完成信息的传递。随着微电子技术的进步，人们成功地研制出了微处理器。微处理器将运算器和控制器两个主要功能部件合二为一，集成到一个芯片里。同时，随着半导体存储器代替磁芯存储器，存储容量成倍的扩大，加上需要计算机处理，加工的信息量与日俱增，以运算器为中心的结构已不能满足计算机发展的需求，甚至会影响计算机的性能。为适应发展的需要，现代计算机组织结构逐步转化为以存储器为中心的组织结构。但是现代计算机基本结构仍然遵循冯·诺依曼思想。

以存储为中心的计算机系统特点是数据的大集中，与主机系统的无关性，可被大量主机设备共用，提供比主机设备本地磁盘更快的IO性能。这种以存储为中心的计算机系统主要组成部分是磁盘阵列设备、FC\IP等主机网络、主机设备

现代计算机运算器的发展主要是在提高CPU的速度，而存储器的发展主要是在扩大容量。现在CPU的处理速度已经远远超过内存的存取速度，相差大概四五个数量级。所以现代计算机的运行速度瓶颈主要是在存储器方面。大概是因为这样的原因所以才说现代计算机是以存储器为中心

两者区别

冯·诺依曼理论的要点是：数字计算机的数制采用二进制；计算机应该按照程序顺序执行。人们把冯诺依曼的这个理论称为冯诺依曼体系结构。从ENIAC到当前最先进的计算机都采用的是冯诺依曼体系结构。所以冯诺依曼是当之无愧的数字计算机之父。

根据冯诺依曼体系结构构成的计算机，必须具有如下功能：把需要的程序和数据送至计算机中；必须具有长期记忆程序、数据、中间结果及最终运算结果的能力；能够完成各种算术、逻辑运算和数据传送等数据加工处理的能力；能够根据需要控制程序走向，并能根据指令控制机器的各部件协调操作；能够按照要求将处理结果输出给用户。

哈佛结构是为了高速数据处理而采用的，因为可以同时读取指令和数据（分开存储的）。大大提高了数据吞吐率，缺点是结构复杂。通用微机指令和数据是混合存储的，结构上简单，成本低。假设是哈佛结构：你就得在电脑安装两块硬盘，一块装程序，一块装数据，内存装两根，一根储存指令，一根存储数据……

是什么结构要看总线结构的。51单片机虽然数据指令存储区是分开的，但总线是分时复用得，所以顶多算改进型的哈佛结构。ARM9虽然是哈佛结构，但是之前的版本也还是冯·诺依曼结构。早期的X86能迅速占有市场，一条很重要的原因，正是靠了冯·诺依曼这种实现简单，成本低的总线结构。处理器虽然外部总线上看是诺依曼结构的，但是由于内部CACHE的存在，因此实际上内部来看已经算是改进型哈佛结构的了。至于优缺点，哈佛结构就是复杂，对外围设备的连接与处理要求高，十分不适合外围存储器的扩展。所以早期通用CPU难以采用这种结构。而单片机，由于内部集成了所需的存储器，所以采用哈佛结构也未尝不可。处理器，依托CACHE的存在，已经很好的将二者统一起来了。

联系冯氏结构的瓶颈，我们很快能想到哈佛结构能够完成指令和数据的并发操作，减轻了程序运行时的访存瓶颈，也就是提高吞吐量，是一种并行结构；而冯氏结构只能是顺序操作，是一种串行的处理方式。原始的冯·诺依曼体系结构，通过引入流水线技术（Pipeline）提高吞吐量。

　　流水线技术与并行处理有所区别，流水线是在顺序指令流计算机中实现处理时间重叠的技术。流水线的并行处理是指完成对一条指令的不同操作（取指令、解码指令、执行指令）的各个部件在时间上是可以同时重叠工作（三级流水线）。CPU是按照取指令、解码指令、执行指令来完成一条指令的操作，当CPU取完第一条指令后，接着解码第一条指令，同时CPU取第二条指令，该步完成后，CPU执行第一条指令，同时解码第二条指令、取第三条指令，如此重叠操作。

　　因此，各部件同时处理是针对不同指令而言的，各部件分别同时为多条指令的不同部分（step、stage）进行工作，以提高各部件的利用率来提高指令的平均执行速度。但是这样虽然提高了系统的速率，由于流水线结构使得不容易计算程序运行的时间，对一些时序要求很严的情况，该结构还是存有弊端的。所以在一些实时性很强的嵌入式系统采用哈佛结构可以高速数据处理，同时读取指令和数据，大大提高了数据吞吐率，保证了系统的可靠性。