

第4讲 冯·诺依曼计算机-程序执行

1、快速浏览---本讲视频都讲了什么？

【视频 4.1 图灵机的思想与模型简介】

计算机是遵循一个理论模型来实现的，这个理论模型就是“图灵机”。**图灵机**是一种用数学方法表达的计算机，它很好地解释了指令、程序与程序执行的概念，阐述了怎样实现“自动计算”。数学上图灵机采用一个五元组来表达程序，采用**状态转换图**来直观地反映程序的执行过程。理解一些“图灵机”的知识对理解程序及程序执行很有帮助……。

【视频 4.2 冯诺依曼计算机思想与构成】

冯·诺依曼计算机是按照图灵机模型实现的计算机。**运算器、控制器、存储器、输入设备和输出设备**，各司其职，相互协作完成程序的执行。这五大部件各具有什么功能呢？为什么要将“运算”和“存储”分离呢？冯·诺依曼计算机体现的是“存储程序”的思想，这种思想的内涵是什么呢，它的价值又体现在哪里呢？--请看视频 4.2。

这一节要理解：这里的**存储器**是指**主存储器**或**内存储器**。**运算器**和**控制器**被集成到一块芯片上，被称为**中央处理单元(CPU)**。

【视频 4.3 自动存取存储器的工作原理】

“存储程序”遇到的第一个问题就是“数据”和“程序”如何实现自动存自动取。在第2讲中我们知道数据和程序都可以被表达成0和1(通过二进制和编码)，那么只要解决了如何自动存取0和1便解决了数据和程序的自动存取问题。存储器在自动存取“数据”和“程序”的时候可以类比“学生宿舍管理”，请看视频是如何类比解释存储器的相关概念的……。

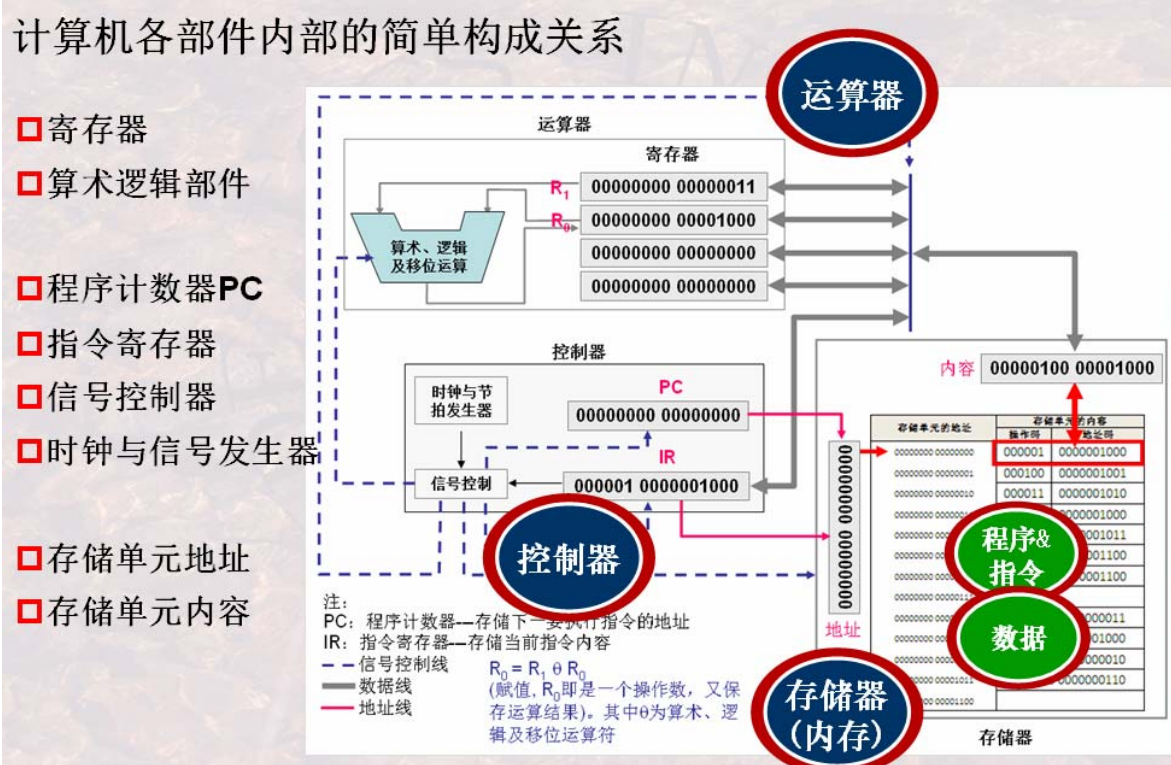
这一节的重点是理解如下几个概念：**存储器、存储单元、地址**。存储器是通过“地址”访问“存储单元”的内容，一个存储单元对应一个地址，即所谓按地址访问。进而要理解**存储容量、地址空间**和**存储字长**等概念。

【视频 4.4 机器指令与机器级程序】

“存储程序”遇到的第二个问题就是“程序”如何来表达？数据和程序都可以通过进位制和编码被表达成0和1串，存储在存储器中。机器能够识别和执行的指令被表达成0和1，用机器指令编写的求解具体问题的程序被表达成0和1。视频中以一个具体的示例展现了什么是**机器级算法、机器指令**和**机器级程序**，以及其如何存储在存储器中。大家可通过读懂这个示例来理解相关的概念。

【视频 4.5 机器级程序的执行机制】

该视频，从概念上给大家一个相对完整的计算机，如下图所示，将在视频 4.6 中用此计算机模拟执行视频 4.4 给出的机器级程序。



观看这段视频，目的是要你理解各个部件内部的一些核心的组件及其功能，以及它们之间是如何连接的，如何协同地完成“存储在存储器中的程序被 CPU 执行”这个任务。以便于依此计算机模拟程序的执行。

【视频 4.6 机器级程序的执行过程模拟】

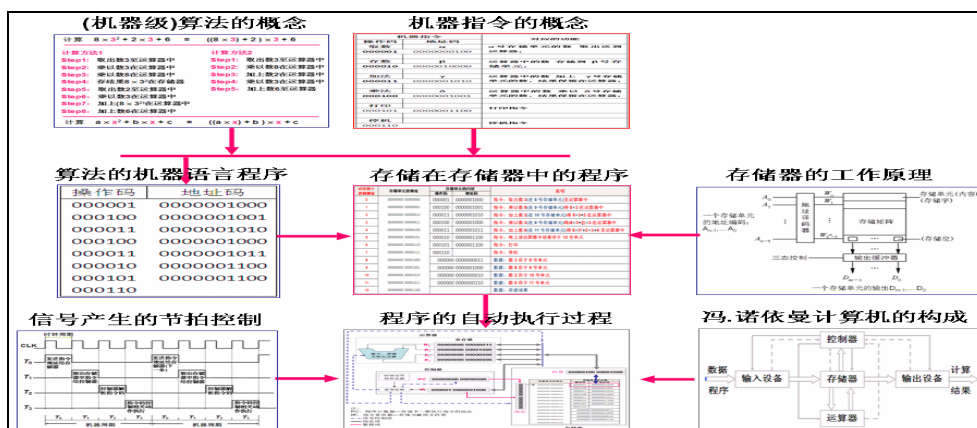
机器级程序的执行，本质上就是产生和发送电信号的过程。本视频以动画的方式，展现了如何执行视频 4.4 给出的机器级程序的过程：读一条指令→执行一条指令→再读下一条指令→再执行下一条指令→...，不断重复此一过程。视频中模拟了两条指令的执行过程及结果，你能否继续后面的指令执行，直到程序执行完毕呢？

本段视频是让大家体验机器级程序的执行过程，其关注点是：**什么时间**(节拍，图中以①②③④⑤来示意，分别表示第 1 个节拍、第 2 个节拍、...) **由哪里向哪里发送了信号**(图中以箭头线示意)，在一个节拍内，可同时发送多个信号。在看此段视频之前一定要看完视频 4.4 和视频 4.5 才可以，否则会造成理解障碍。

2、学习要点指南

2.1 要点一：理解“机器如何执行存储在内存中的程序”

本讲内容概述起来就是让大家理解下面这张图及其所表达的“**算法程序化**→**程序指令化**→**指令存储化**→**执行信号化**”思维模式。



计算机就是能够进行问题求解的机器，问题求解的关键是构造可让计算机执行的算法。算法可在不同层面上构造，例如可在高级语言层面构造，也可在机器语言层面构造等。

本讲首先通过示例，让大家理解机器级算法是什么样的，进一步给大家展现了如何将机器级算法表达为机器可以执行的程序-即机器级程序，为此给大家介绍了机器能够识别和执行什么---机器指令，以及用机器指令编写的机器级程序。

当编制好机器级程序后，需要将程序和数据存储在存储器中，为此需要理解存储器的工作原理，理解存储单元和存储单元地址的概念，展现了数据和程序是如何存储在存储器中。

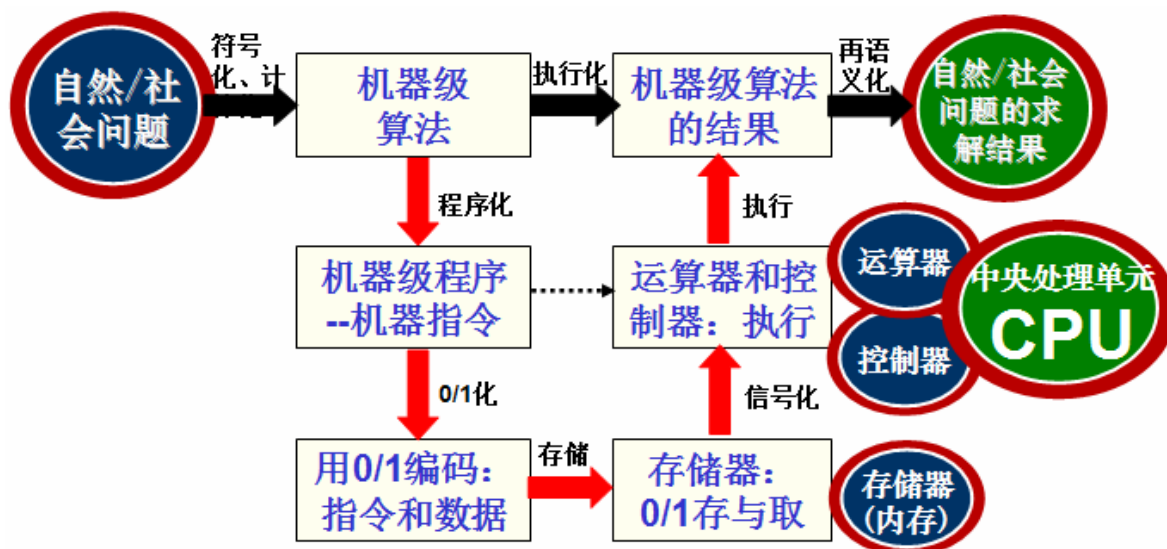
进一步，又解释了冯·诺依曼计算机的五大部件**运算器**、**控制器**、**存储器**、**输入设备**和**输出设备**及其连接方式，各部件内部的核心组件，以示意计算机内部就是元器件及信号连接线，所谓的执行，就是信号的产生、发送和控制，而控制不同信号传递关系(暨什么时间给谁发出信号)是通过时钟和节拍来实现的。

当上述内容理解后，就是**要模拟计算机来执行一个机器级程序**。当你知道了机器是如何执行程序的时候，则你必然会理解如何编写程序。很多人不会编写程序，对一些计算机语言的要素理解不透，其根本原因是他对计算机如何执行程序不理解。

这就是“**算法程序化**→**程序指令化**→**指令存储化**→**执行信号化**”思维模式，它是可实现的思维模式，但并未更多涉及实现的细节。这种思路，进一步展现出来如下图所示：

自然/社会问题求解过程是通过符号化、计算化表达为机器级算法，用计算机执行这个算法得到结果，再语义化还原回自然/社会问题层面得到求解结果，这是黑色箭头所指示的过程。但这一过程，实际由计算机内部执行的时候，则由红色箭头所示：将机器级算法表达为机器级程序，进一步机器级程序和数据被表达为 0 和 1，存储于存储器中，由 CPU 的运算器和控制器读取存储器

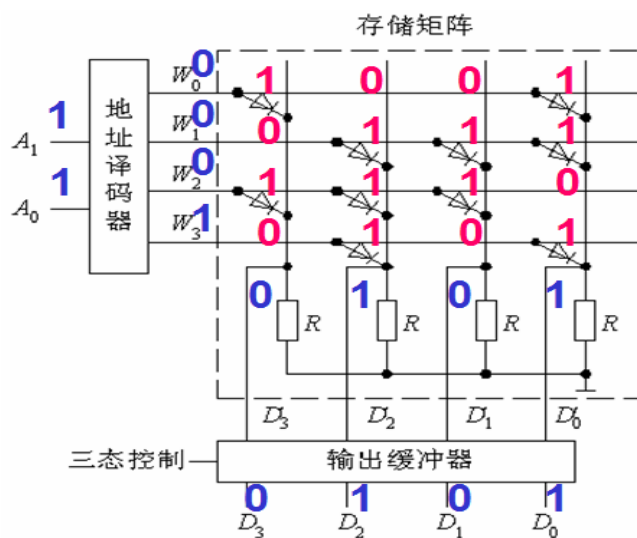
中内容获得程序和数据并予以执行，才能得到机器级算法的结果。红色箭头指示的就是计算机内部执行算法的机理。



本讲内容是对第一讲“计算之树”中的“冯·诺依曼机”思维的深化讲授。

2.2 要点二：存储器的存取机理

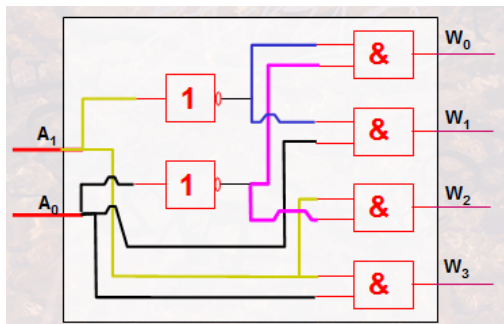
存储器的存储机理可以通过下图来理解。如何理解该图呢？



可以如下理解该图：

(1)地址译码器。地址译码器将两位的地址编码 A_1A_0 翻译成四条地址线 $W_3W_2W_1W_0$ 的某一条有效，即 $W_3W_2W_1W_0$ 中只能有一个为 0。例如 $A_1A_0 = 01$ ，则 $W_3W_2W_1W_0 = 0010$ ；如 $A_1A_0 = 11$ ，则 $W_3W_2W_1W_0 = 1000$ 。现在如上图 $A_1A_0 = 11$ ，则只有 W_3 为 1，其余全为 0。

此译码器可用“与”门和“非”门如下图连接实现(此图在第2讲视频2.5中有介绍), **如果不考虑硬件因素, 这种译码本质就是一组逻辑运算, 例如: $W_0 = (\text{NOT } A_1) \text{ AND } (\text{NOT } A_0)$ 。**



(2)再看存储矩阵部分。存储矩阵可被认为是 W 线(水平线)和 D 线(垂直线)是否相交所构成, 有二极管连接的为相交, 表示存储的是 1; 无二极管连接的为不相交, 表示存储的是 0。有二极管连接的, 假设 W_i 线和 D_j 线有二极管连接, 则当 W_i 线为高电平时, 二极管导通, 则 D_j 线为高电平, 当 W_i 线为低电平时, 二极管截止, 则不会改变 D_j 线的电压。由此 W_i 为高电平时控制着 D_j , 而 D_j 为高电平(1)还是低电平(0)则取决于是否与 W_i 有二极管连接。

一行是一个**四位**的存储单元, 由一条 W 线控制。一个存储单元的位数, 被称为**存储字长**。

存储矩阵中显示有**四个**存储单元, 四列。这四个存储单元分别存储的是 1001, 0111, 1110, 0101 (四行红色的数字)。存储单元的个数多少, 一取决于表达存储单元地址的编码位数, 二取决于存储矩阵中实际存在的存储单元, 此即存储空间。位数越多, 存储空间越大。例如 20 位的 0/1 编码位数, 则有 2^{20} 个存储单元。要注意, 20 位的编码位数仅说明其可管理多达 2^{20} 个存储单元, 但其是否真有这么多的存储单元, 取决于实际的存储矩阵。

(3)进一步看给定一个地址, 是否能正确地读出一行的数据呢?

假设给定 $A_1A_0 = 11$, 由译码器能够确认 $W_3W_2W_1W_0 = 1000$ 。即只有 W_3 为 1, 其余全为 0。我们现在看 D 线: 先看 D_3 , 由于 W_0, W_1, W_2 均为 0, 因此不管其与 D_3 有无二极管连接, 不改变 D_3 的电压(D 线初始都是接地的); 由于 W_3 为 1, 但因其和 D_3 没有二极管连接, 所以 D_3 为低电平 (即输出 0)。再看 D_2 , 由于 W_0, W_1, W_2 均为 0, 因此不管其与 D_2 有无二极管连接, 对 D_2 都无影响; 由于 W_3 为 1, 但因其和 D_2 有二极管连接, 所以 D_2 为高电平 (即输出 1)。类似的可以确定 $D_3D_2D_1D_0 = 0101$ 。

(4)大家可以进一步验证给定任意一组 A_1A_0 值都能正确地读出相应的存储单元的值。

(5)总结一下:上图做了两个变换, 一个是将 A_1A_0 转换成 $W_3W_2W_1W_0$, 另一个是将 $W_3W_2W_1W_0$ 转换成 $D_3D_2D_1D_0$, 后者依据有无二极管连接进行转换。

根本来讲, 这些转换都是通过基本的逻辑运算: 与、或、非运算来实现的。如下所示。

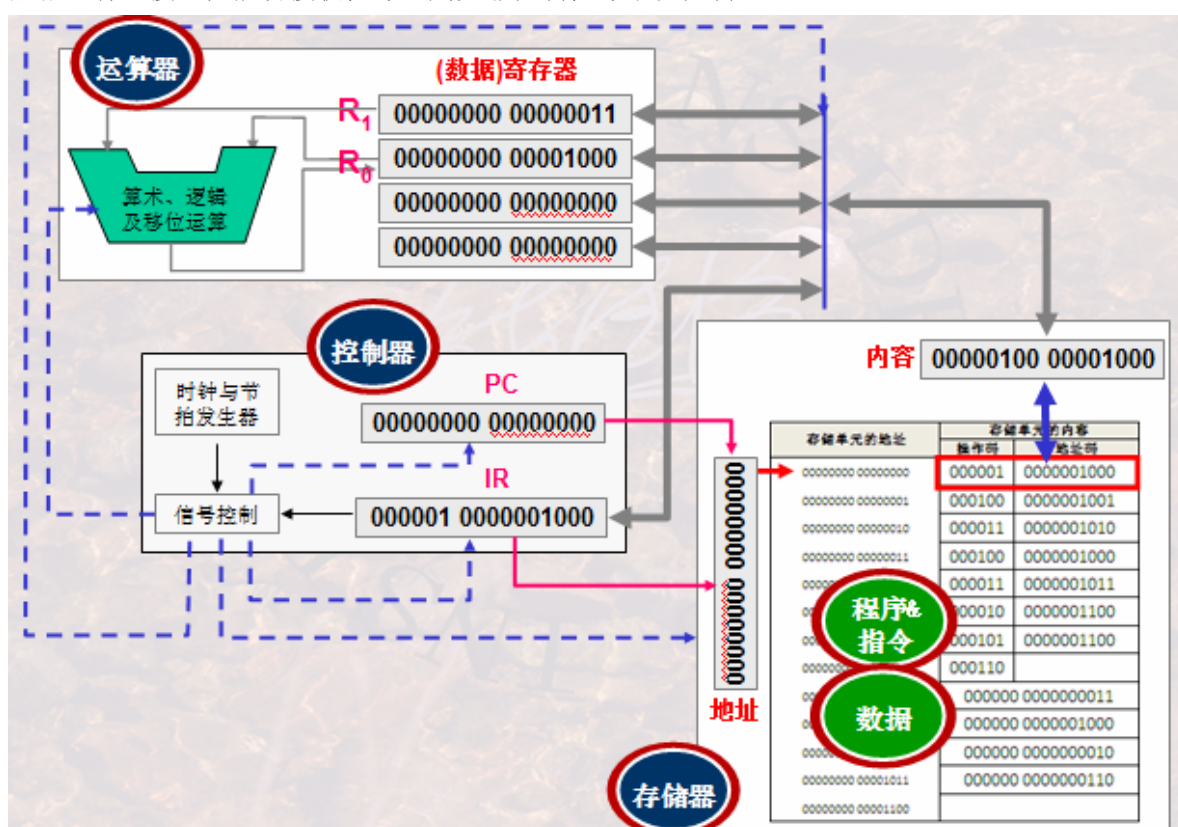
$W_0 = (\text{NOT } A_0) \text{ AND } (\text{NOT } A_1)$	$D_3 = W_0 \text{ OR } W_2$
$W_1 = A_0 \text{ AND } (\text{NOT } A_1)$	$D_2 = W_1 \text{ OR } W_2 \text{ OR } W_3$
$W_2 = (\text{NOT } A_0) \text{ AND } A_1$	$D_1 = W_1 \text{ OR } W_2$
$W_3 = A_0 \text{ AND } A_1$	$D_0 = W_0 \text{ OR } W_1 \text{ OR } W_3$

2.3 要点三：冯·诺依曼计算机器

(1)冯·诺依曼计算机的组成：**运算器**、**控制器**、**存储器**、**输入设备**和**输出设备**。要理解这五大部件的基本功能。其中，**CPU**，中央处理单元，是将**运算器**和**控制器**集成在一块芯片上所形成的，是基本的程序执行机构。**存储器**，这里指的是**内存**，又称**主存**，是可按地址访问存储单元内容的部件。

这些部件的具体实现细节可不考虑，但其内部的核心组件应该了解：**运算器**中有一些但不是很多的**寄存器**，有能够执行基本运算的**算术逻辑部件**。**控制器**中有一个**程序计数器 PC**、一个**指令寄存器 IR**、一个**时钟/节拍发生器**，还有一个**控制信号发生器**。**存储器**有一个**地址寄存器**和**内容寄存器**(或称**内容缓冲器**)。

这些部件/组件之间都有连接线。如下图示意，**存储器**的**内容寄存器**与**控制器**的**指令寄存器 IR**有连接，与**运算器**的 4 个**寄存器**也有连接，那么从存储器中取出的内容是传给谁呢？这时候就要看**控制器**的**控制信号发生器**发出的控制信号：如果控制信号发给指令寄存器，则指令寄存器接收存储器的内容；如果控制信号发给运算器的某一个数据寄存器，则该寄存器接收存储器的内容；如果控制信号同时发给两个数据寄存器，则此两个数据寄存器可同时接收存储器的内容。因此虽然相互有连接，但能否接收信号，则要靠控制信号予以控制。



这些部件和组件的理解，是你能否理解“计算机”的关键，这里不要求你理解更深入的东西，但要求你能从思维的角度而不是从实现的细节角度理解它。

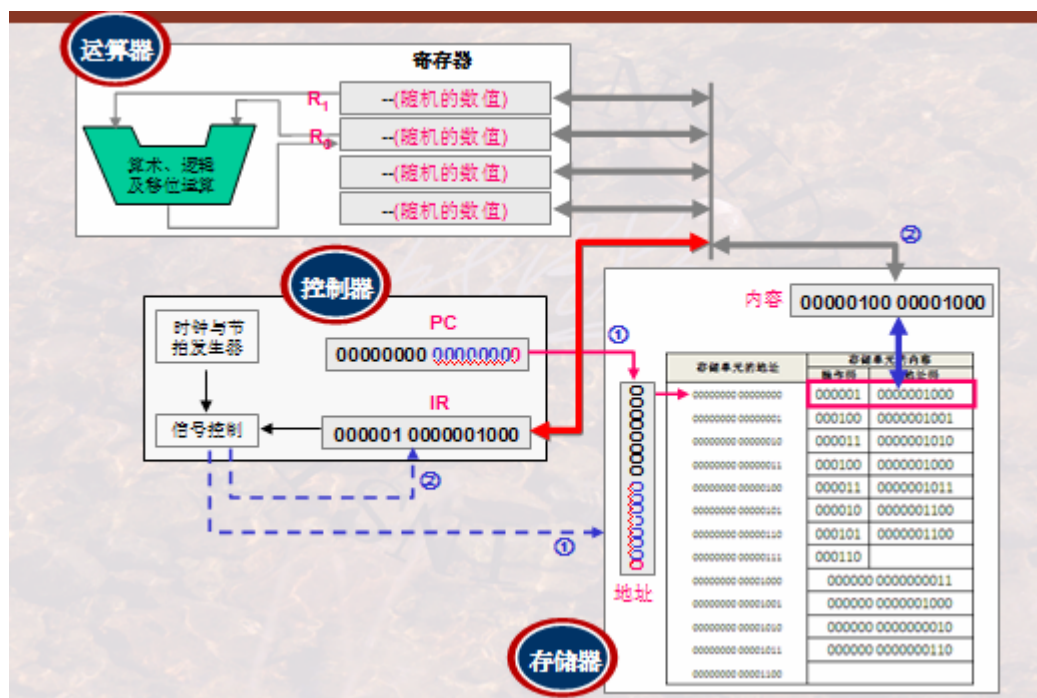
2.4 要点四：冯·诺依曼计算机执行程序的过程

理解计算机执行程序的过程，就是要理解“执行信号化”，所谓执行一条指令，即是依据该指令的要求，发出相应的信号给各个部件。信号发生的时候，有可能有冲突，此时要靠时钟和节拍来控制，即第1个节拍发出什么信号，第2个节拍发出什么信号，...

我们来看读一条指令的过程：第1个节拍①内发地址信息给存储器，将PC中的值送给存储器的地址寄存器，同时发一个控制信号给存储器，通知存储器开始工作；第2个节拍②内存储器开始工作，依据地址寄存器的地址编码，找到相应的存储单元，将该存储单元的内容送到内容寄存器中，同时发一个控制信号给IR寄存器，通知IR接收内容寄存器中的数据。如此将存储器中的指令读到了控制器的IR中。

后续指令的读取与执行过程可类似完成。

这里的关注点是：**什么时间**(节拍，图中以①②③④⑤来示意，分别表示第1个节拍、第2个节拍、...) **由哪里向哪里发送了信号**(图中以箭头线示意)，在一个节拍内，可同时发送多个信号。



大家可以试着完整地将存储器中的程序执行完，则对理解机器级程序一定是有帮助的。

3、常见问题

3.1 存储器应该理解到什么程度？应把哪些概念理解清楚？

对于存储器，下列概念应理解清楚：

存储器、存储单元、存储单元的地址(简称存储地址)、存储空间、存储字长等；

理解清楚下列内容：

- (1) (内或主) 存储器是按地址访问的存储器，一个地址编码对应一个存储单元；
- (2) 地址编码的位数，或者说地址线的位数，决定了其能访问的存储单元的个数。例如 n 位地址线，或者说 n 位的地址编码，能够访问 2^n 个存储单元。
- (3) 一个存储单元可以有 m 位，称存储字长为 m 。

上述内容对于计算机语言或者程序设计的学习者来说同样是很重要的内容，比如“变量”就是一个或多个存储单元，取决于变量的类型；比如“指针”，则是存放地址的一个变量，即该变量的内容是指向所要找内容的地址等等。

3.2 冯·诺依曼计算机应该理解到什么程度？应把哪写概念理解清楚？

对于冯·诺伊曼机器，下列概念应理解清楚：

CPU、运算器、控制器、存储器、输入设备及输出设备；

机器级算法、机器指令与机器级程序；

数据寄存器、指令寄存器、程序计数器；

理解清楚下列内容：

- (1) 存储程序的基本思想；
- (2) 指令执行的过程，即是在时钟/节拍控制下发出一系列信号的过程，包括控制信号、数据信号和地址信号等；
- (3) 机器级程序的执行过程就是读指令→执行指令→再读指令→再执行指令→... →一直到结束的过程。