2.2 图灵机与冯·诺依曼计算机

哈尔滨工业大学（深圳）
计算机学院

图灵机与冯·诺依曼计算机

一、图灵机

二、冯.诺伊曼计算机的功能与构成

三、存储器的基本概念

四、自动存取：存储器的工作原理

五、存储器容量不够怎么办?

六、机器指令与机器程序

七、一台典型的计算机

八、指令执行

九、机器程序的执行过程模拟

图灵机：图灵

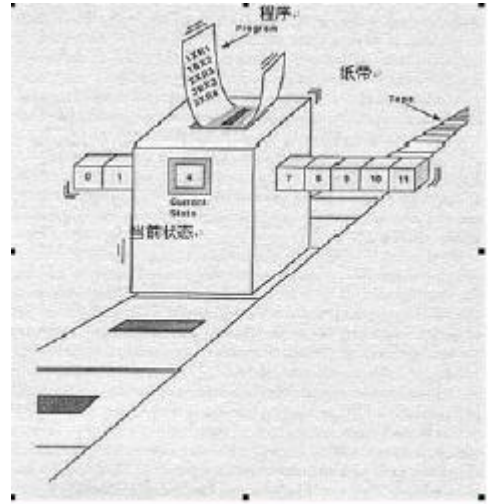
图灵及其贡献

◆图灵(Alan Turing, 1912~1954)，出生于英国伦敦，19 岁入剑桥皇家学院，22 岁当选为皇家学会会员。

◆1937 年，发表了论文《论可计算数及其在判定问题中的应用》，提出了**图灵机模型**，后来，冯·诺依曼根据这个模型设计出历史上第一台电子计算机。

◆1950 年，发表了划时代的文章：《机器能思考吗?》,成为了人工智能的开山之作。

◆计算机界于1966年设立了最高荣誉奖：**ACM图灵奖**。



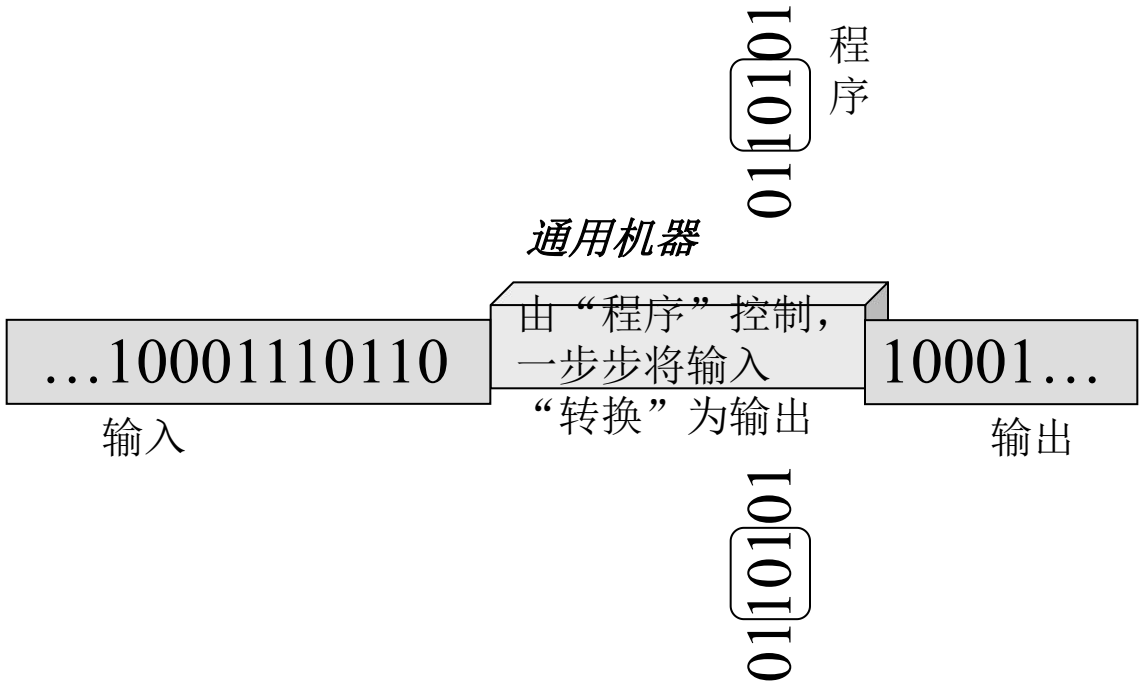
图灵机装置示意图

你能查阅一下哪些人获得图灵奖了吗？因为什么贡献而获奖呢？

图灵机：图灵认为什么是计算

什么是计算

◆所谓计算就是计算者(人或机器)对一条两端可无限延长的纸带上的一串0或1，执行指令一步一步地改变纸带上的0或1，经过有限步骤最后得到一个满足预先规定的符号串的变换过程。



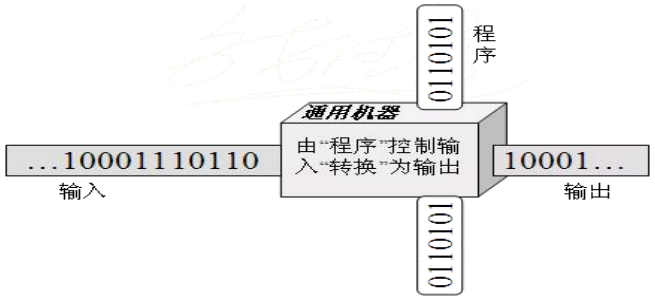
图灵机：图灵认为什么是计算

图灵机的思想

是关于数据、指令、程序及程序/指令自动执行的基本思想。

- 输入被制成一串0和1的纸带，送入机器中----数据。如000100001000011…
- 机器可对输入纸带执行的基本动作包括：“翻转0为1”，或“翻转1为0”，“前移一位”，“停止”。
- 对基本动作的控制----指令，机器是按照指令的控制选择执行哪一个动作，指令也可以用0和1来表示：**01**表示“翻转0为1”(当输入为1时不变)，**10**表示“翻转1为0”(当输入0时不变)，**11**表示“前移一位”，**00**表示“停止”。
- 输入如何变为输出的控制可以用指令编写一个程序来完成，如：01111011011011011100…
- 机器能够读取程序，按程序中的指令顺序读取指令，

读一条指令执行一条指令。由此实现自动计算。



图灵机：图灵机模型

图灵机模型

◆基本的图灵机模型为一个七元组,如右图

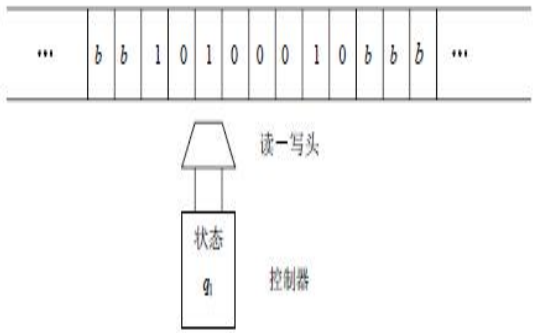
◆几点结论:

(1) 图灵机是一种思想模型，它由一个控制器(有限状态转换器)，一条可无限延伸的带子和一个在带子上左右移动的读写头构成。

(2) 程序是五元组<q,X,Y,R(或L或N),p>形式的指令集。其定义了机器在一个特定状态q下从方格中读入一个特定字符X时所采取的动作为在该方格中写入符号Y, 然后向右移一格R (或向左移一格L或不移动N), 同时将机器状态设为p供下一条指令使用。

$M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$

其中:



Q:状态的有穷集合

q_0:开始状态

F:终止状态集合

Γ:带符号表

B:空白符号

Σ:输入字母表

δ:移动函数,(1):$\delta(q,X)=(p,Y,R)$表示M在状态q读入符号X,将状态改为p,并在这个X所在的带方格中印刷符号Y,然后将读头向右移动一格.

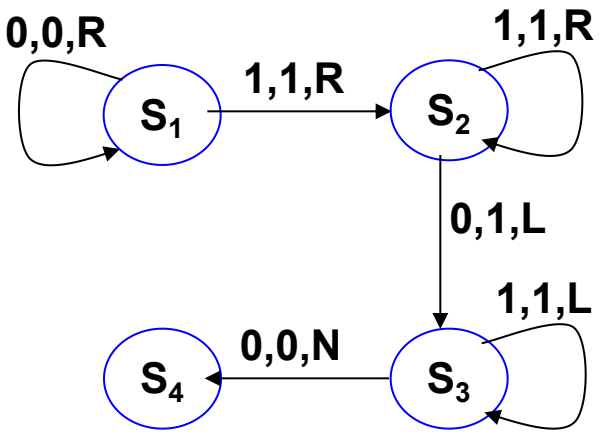
(2):$\delta(q,X)=(p,Y,L)$表示M在状态q读入符号X,将状态改为p,并在这个X所在的带方格中印刷符号Y,然后将读头向左移动一格.

图灵机：图灵机模型

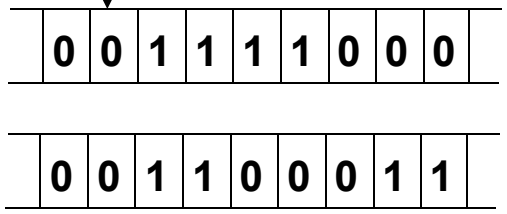
图灵机模型示例。(注:(q,X,Y,R(或L或N),p), 状态图中圆圈内的是状态，箭线上的是<X,Y,R>, 其含义见前页)

S_1：开始状态
S_2：右移状态
S_3：左移状态
S_4：停机状态

$(S_1,0,0,R,S_1)$
$(S_1,1,1,R,S_2)$
$(S_2,1,1,R,S_2)$
$(S_2,0,1,L,S_3)$
$(S_3,1,1,L,S_3)$
$(S_3,0,0,N,S_4)$

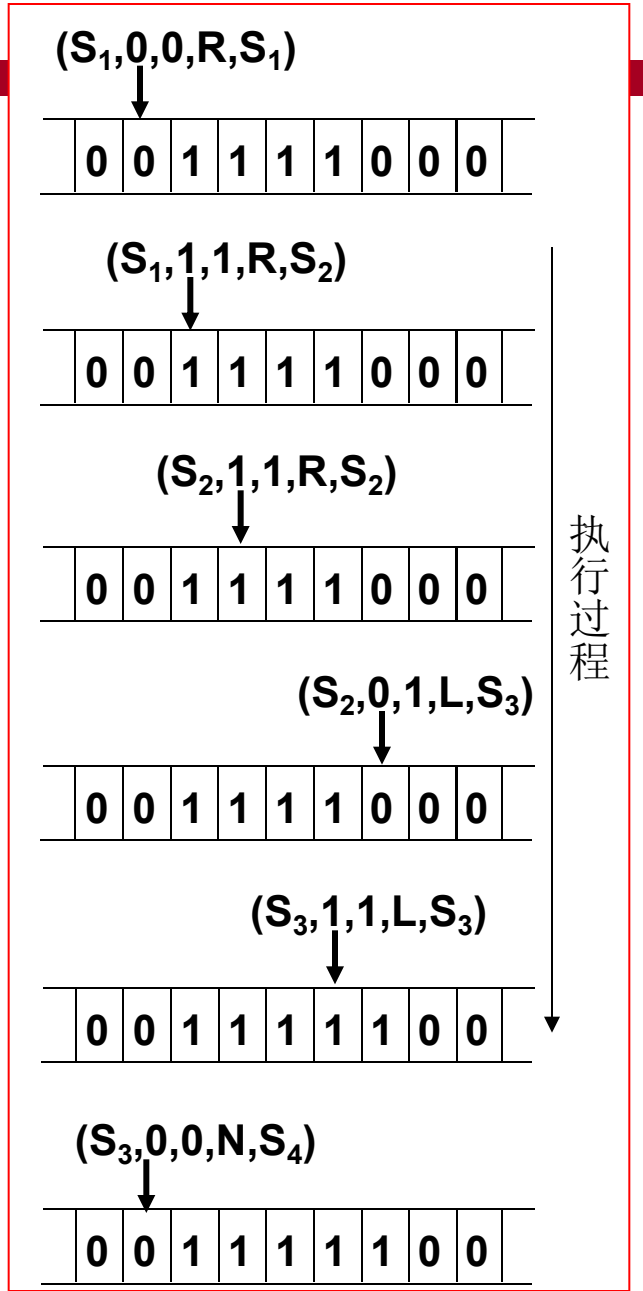


控制器



你能否用另一个输入模拟一下这个程序的执行呢?

功能：将一串连续1的后面再加一位1



图灵机：图灵机模型

几点结论(续):

◆(3)图灵机模型被认为是计算机的基本理论模型

----计算机是使用相应的程序来完成任何设定好的任务。图灵机是一种离散的、有穷的、构造性的问题求解思路，一个问题的求解可以通过构造其图灵机(即程序)来解决。

◆(4)图灵认为：凡是能用算法方法解决的问题也一定能用图灵机解决；凡是图灵机解决不了的问题任何算法也解决不了----图灵可计算性问题。

图灵机：计算机的理论模型

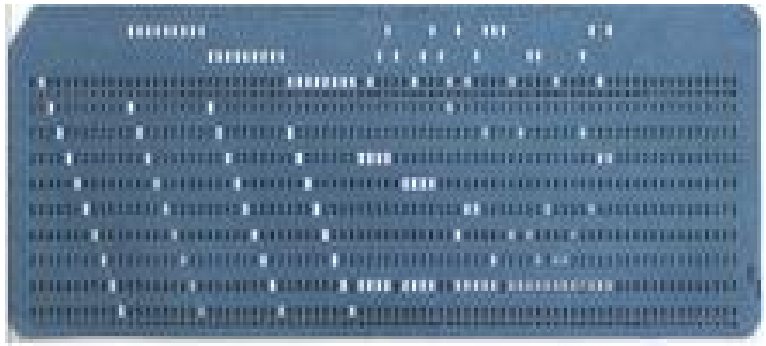
输入/输出都是0和1的形式表达

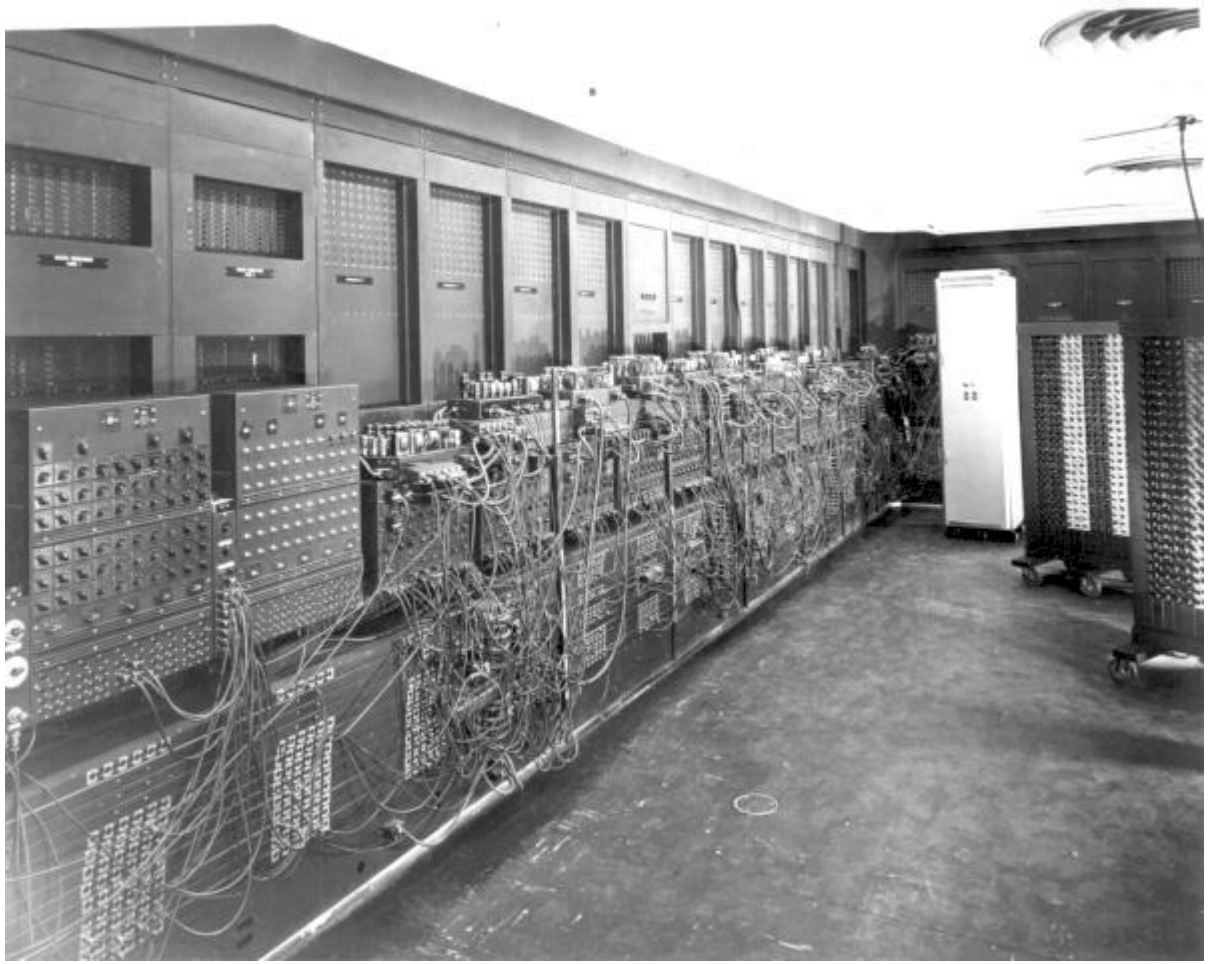
程序和指令也是0和1的形式表达

程序可用状态转换图来表达

冯.诺伊曼计算机的功能与构成

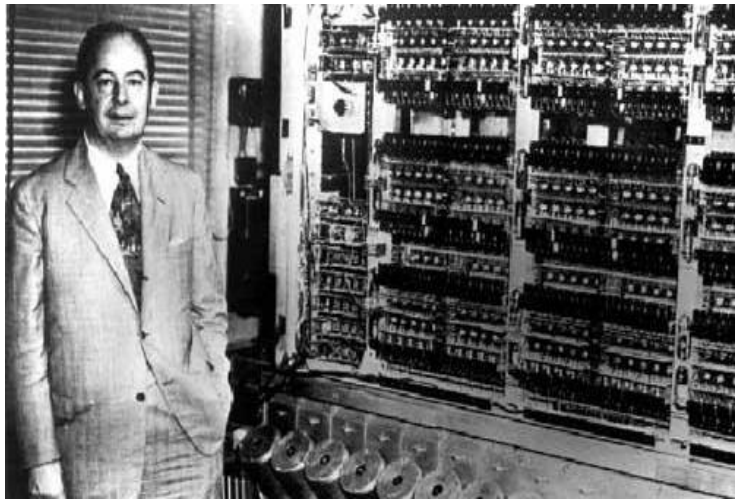
最早的计算机ENIAC





冯.诺伊曼计算机的功能与构成

冯.诺依曼(Von.Neumann)计算机



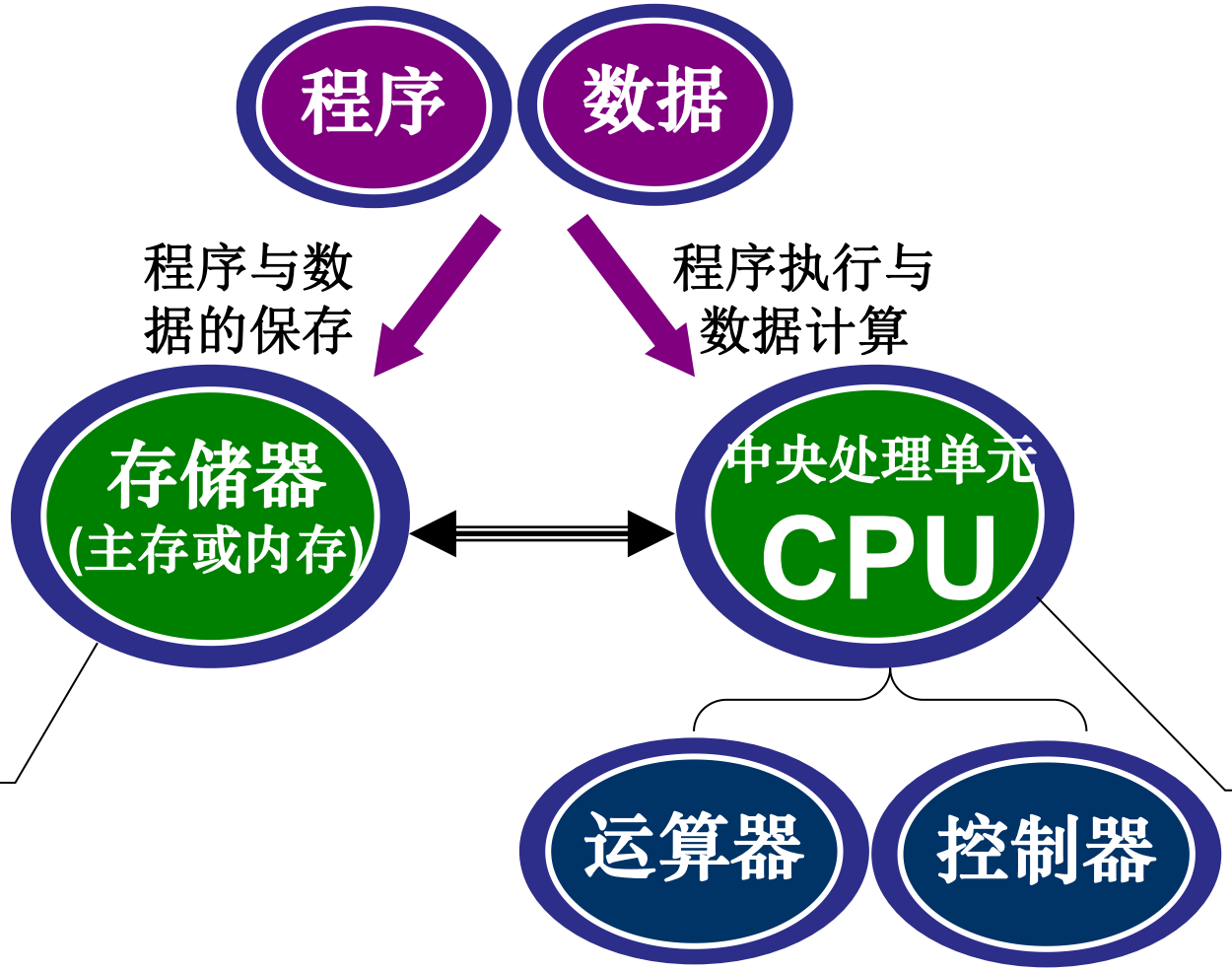
◆1944~1945年间，冯.诺伊曼提出“存储程序”的计算机设计思想，并进行了实践，现代计算机普遍来讲属于冯.诺伊曼机体系。

◆冯.诺伊曼机的基本思想：

- **运算**和**存储**分离
- **存储程序**：指令和数据以同等地位事先存于存储器，可按地址寻访，连续自动执行。
- 五大部件构成：**运算器**、**控制器**、**存储器**、**输入设备**和**输出设备**
- 指令和数据用**二进制**表示，指令由操作码和地址码组成
- 以运算器为中心，控制器负责解释指令，运算器负责执行指令

冯.诺伊曼计算机的功能与构成

冯.诺伊曼计算机器的核心



以【存储单元】为单位进行存储
一个存储单元可以是8位、16位、32位或64位
按【地址(编号)】访问存储单元

以【指令】为单位进行读取并执行。
一条指令可以保存在1个或多个存储单元中。
基本的运算指令就是两个数的加、减、乘、除等运算。

单选题 1分

设置

13

关于“存储程序”，下列说法不正确的是_____。

A 将“指令”和“数据”以同等地位保存在存储器中，以便于机器自动读取自动处理；

B 之所以将“程序”和“数据”事先存储于存储器中，是因为输入的速度满足不了机器处理的速度，为使机器连续自动处理，所以要“存储程序”；

C 依据“存储程序”原理，机器可由四大部分构成：运算器、存储器、输入设备和输出设备；

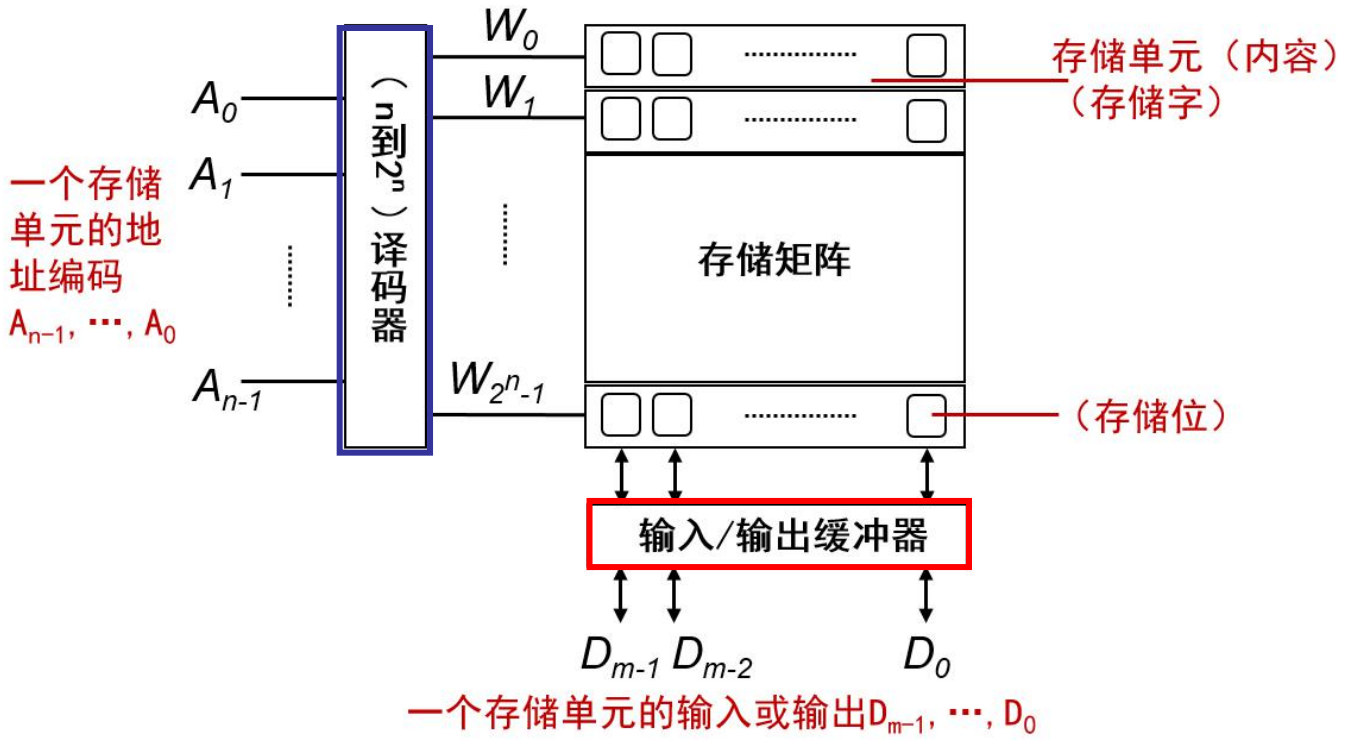
D 冯.诺依曼计算机的本质就是“存储程序、连续自动执行”。

提交

存储器的基本概念

什么是存储器?

- 存储器:能够按地址读或写每一个存储单元的部件。
- 地址：是一个n位的0/1编码，每一个编码指向一个存储单元，通常记为$A_{n-1}\cdots A_0$。
- 存储单元：一个存储单位可以保存一个m位的数据，通常记为$D_{m-1}\cdots D_0$。



存储器的基本概念

存储器的容量

地址编码n位： $A_{n-1}\cdots A_0$

地址空间：

00000000…00000000
00000000…00000001
… …
11111111…11111111

} 2^n个存储单元

存储字长m位：$D_{m-1}\cdots D_0$

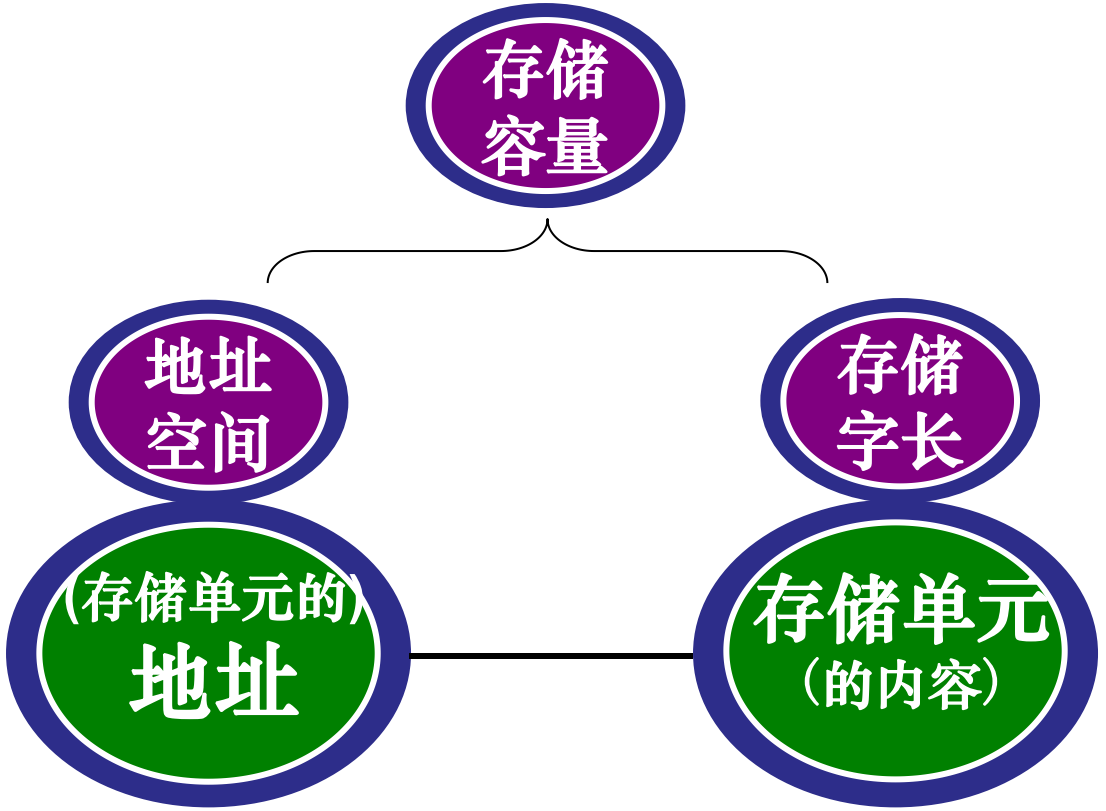
每个存储单元都是m位

存储容量 = $2^n \times m$ Bit(位)

= $2^n \times m/8$ Byte（字节）

存储容量的单位：2^{10}为换算单位

DB、NB、BB、YB、ZB、EB、PB、TB、GB、MB、KB、Byte



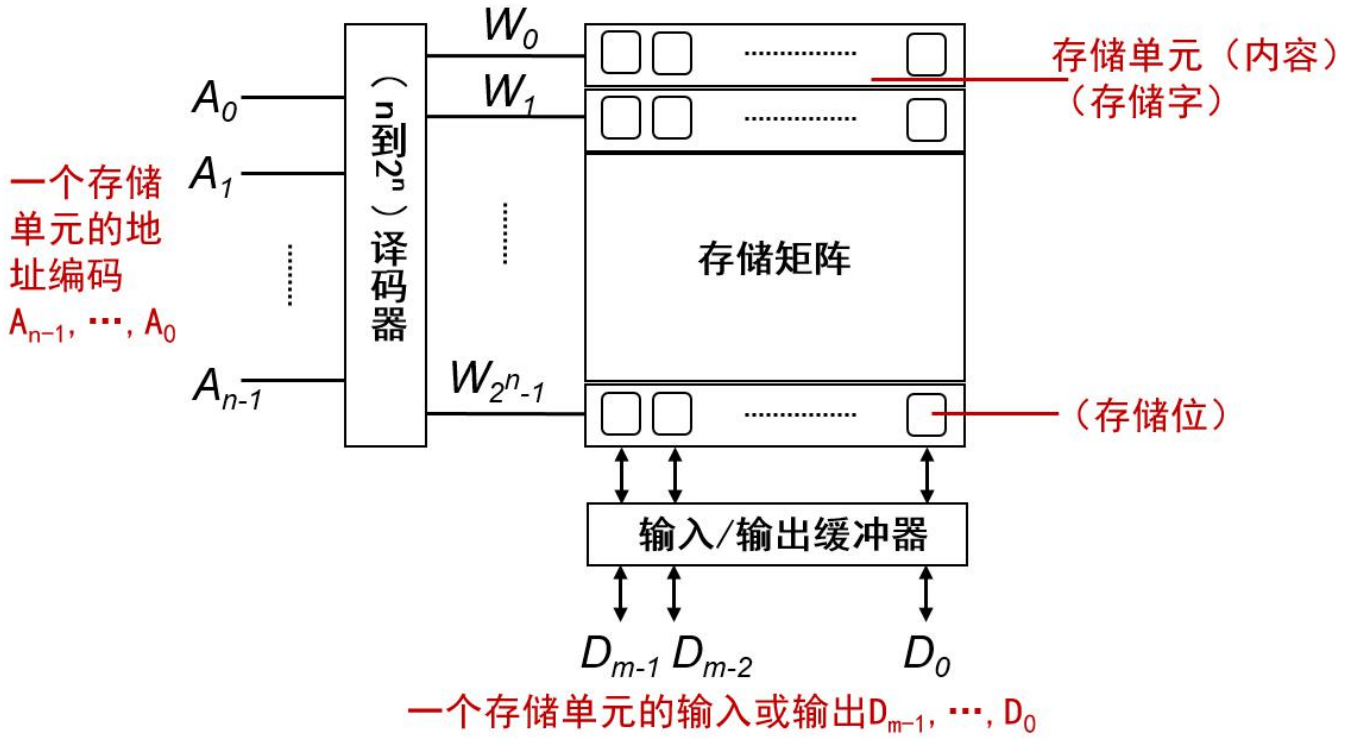
自动存取：存储器的工作原理

对比宿舍楼，理解存储器的相关概念

概念映射

存储器	宿舍楼
存储单元	房间
存储位(存0或存1)	床位(住人/不住人)
地址编码$A_{n-1}…A_0$	房间号
单元控制线W_i	房间钥匙
输出缓冲器	公共的走廊及大门
…	……

从存储器与宿舍楼的概念对比中，你能发现什么异同吗？



自动存取：只读存储器（ROM）的工作原理

存储器是怎样存储0和1的？又是怎样控制存取的？

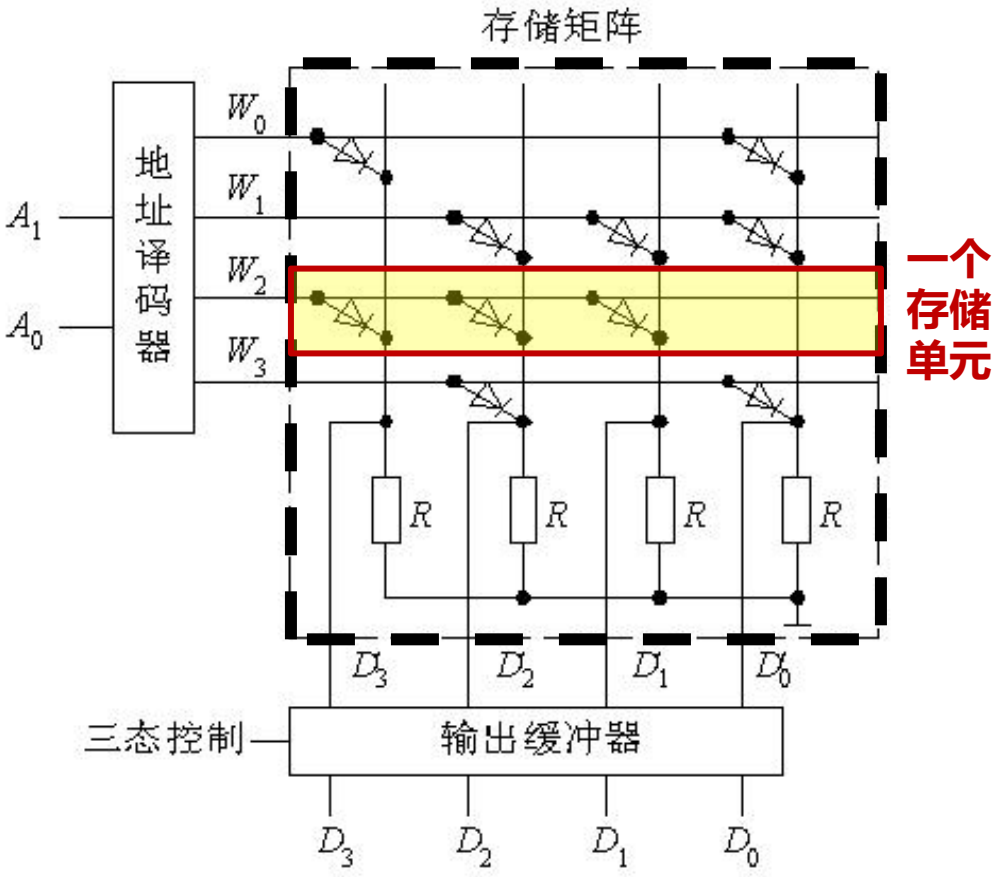
存储器内部的实现示例

◆当单元控制线W和数据线D间连接有二极管时，则存储的是1，否则，存储的是0。这是只读存储器（ROM，只能读出不能写入）示例。

■当单元控制线W和数据线D间连接有二极管时，由单元控制线决定其是输出1或0，即：当单元控制线为1（高电压）时，则输出1（高电压），而当单元控制线为0（低电压）时，则输出0（低电压）。没有连接的，则不受单元控制线影响，始终输出0（低电压）。

■W_3,W_2,W_1,W_0随着A_1A_0的值同时只能有一个为1（高电压）其它为0（低电压），即控制一个存储单元所有位的读写。

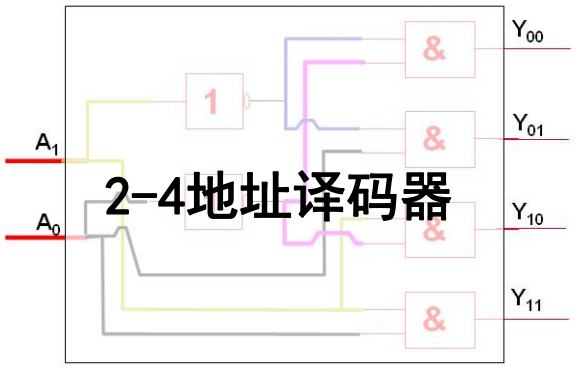
■尽管所有存储单元的第i位都连接到D_i，但只有将读取存储单元的第i位对D_i产生作用。



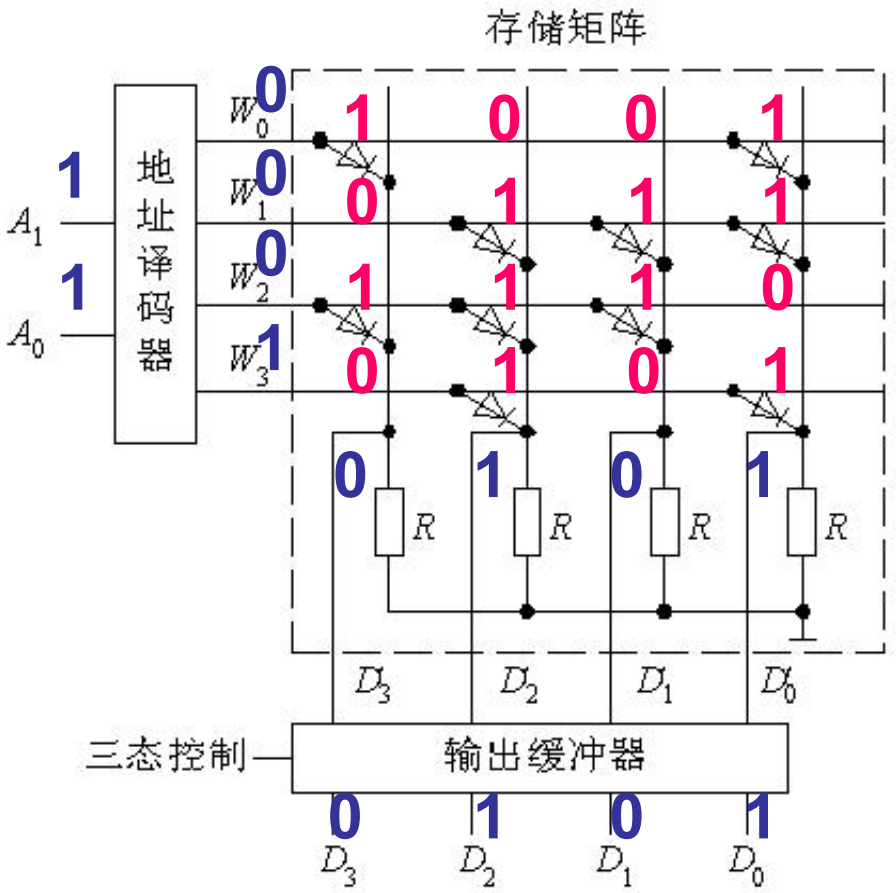
二极管ROM结构示例
(2位地址控制4个存储单元, 每个存储单元是4位0/1)

自动存取：存储器的工作原理

读出过程示例：按地址读取存储单元的内容



将地址编码转换为地址单元控制信号
类比:将房间号转换成房间钥匙



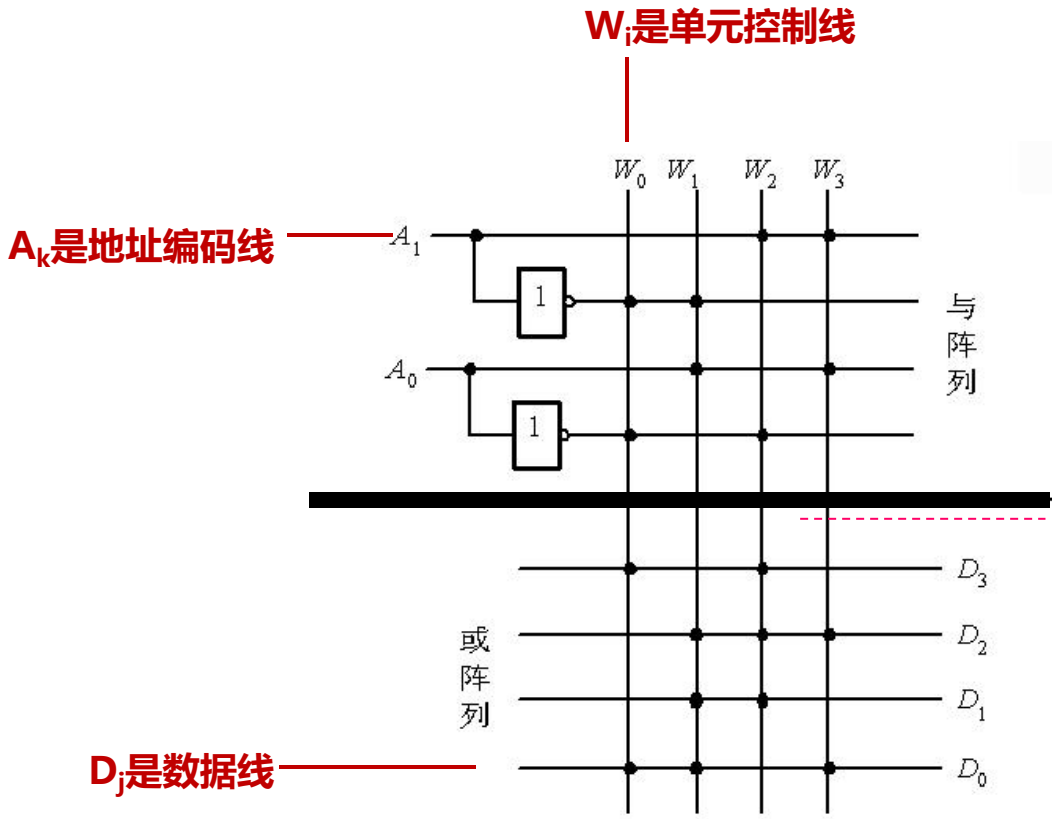
二极管ROM结构示例
(2位地址控制4个信息单元, 每个信息单元是4位0/1码)

存储器实现的逻辑控制

存储矩阵的逻辑控制关系

存储矩阵：一种【与或逻辑】阵列

- 【地址编码线】与【单元控制线】有黑点则连接，无黑点则不连接。
- 【单元控制线】与【数据线】有黑点则连接，无黑点不连接。
- 高/低电压信号，即0,1，通过连接点相互传递。
- 上半部是【与】阵列，下半部是【或】阵列。



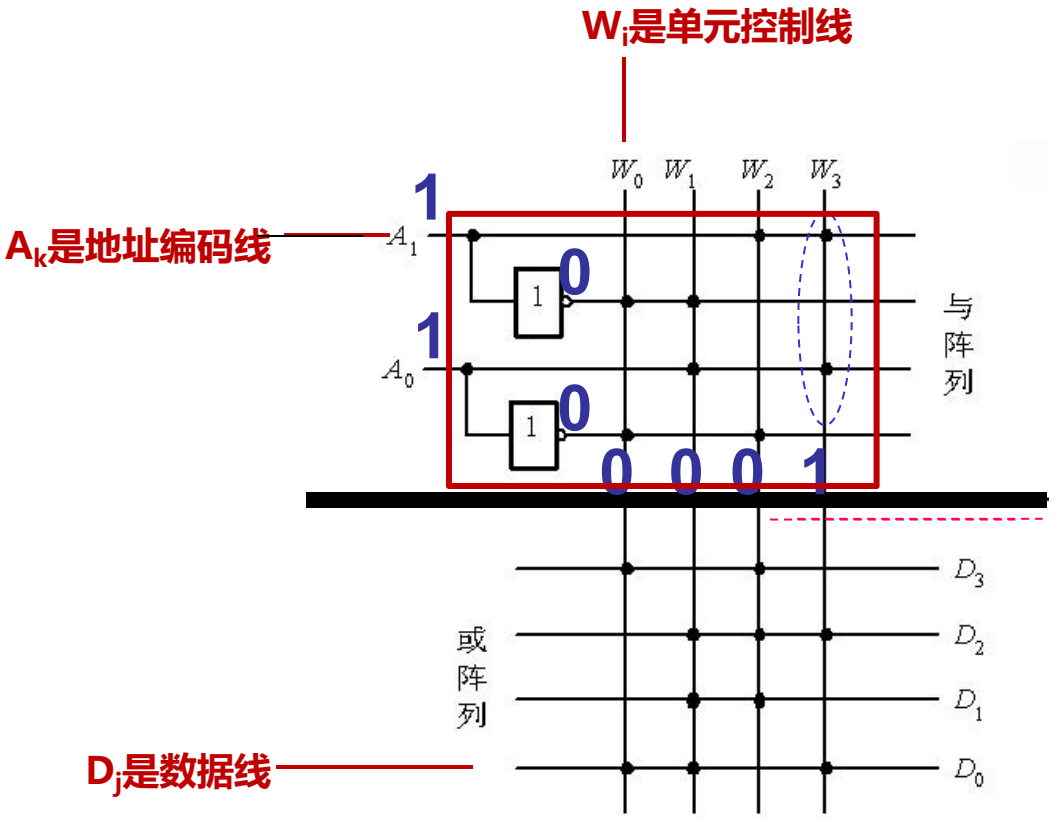
存储器实现的逻辑控制

存储矩阵的逻辑控制关系

译码器：将A_1A_0地址编码翻译到只有一条单元控制线W_i为1，其它为0。

- 【与】阵列，表述了如何由A_1A_0的值产生W_0,W_1,W_2,W_3的值（只能有一条线为1），由横线向纵线传输信号—上半部。
- 同一【单元控制线】上各连接点之间是“与”关系。即只有各连接点都为1时，该单元控制线的信号为1，否则为0。

$W_0 = (\text{NOT } A_0) \text{ AND } (\text{NOT } A_1)$
$W_1 = A_0 \text{ AND } (\text{NOT } A_1)$
$W_2 = (\text{NOT } A_0) \text{ AND } A_1$
$W_3 = A_0 \text{ AND } A_1$



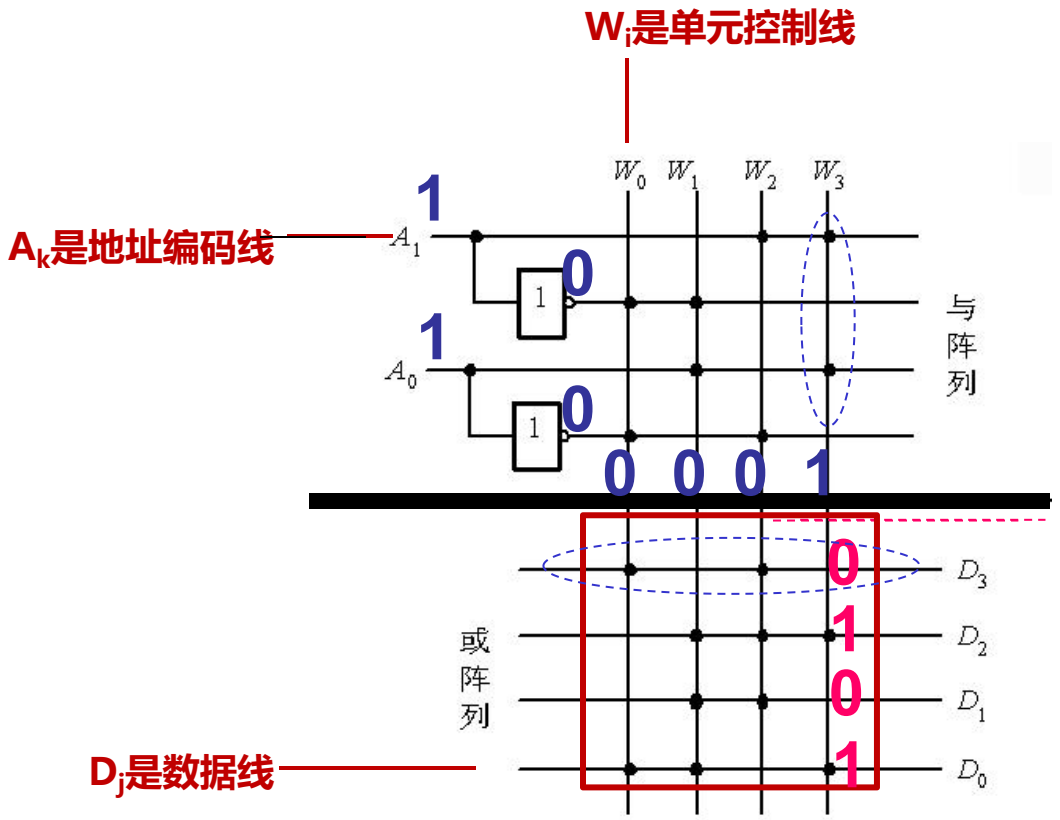
存储器实现的逻辑控制

存储矩阵的逻辑控制关系

输出矩阵：将W_i为1的单元控制线控制的存储单元的值进行输出。

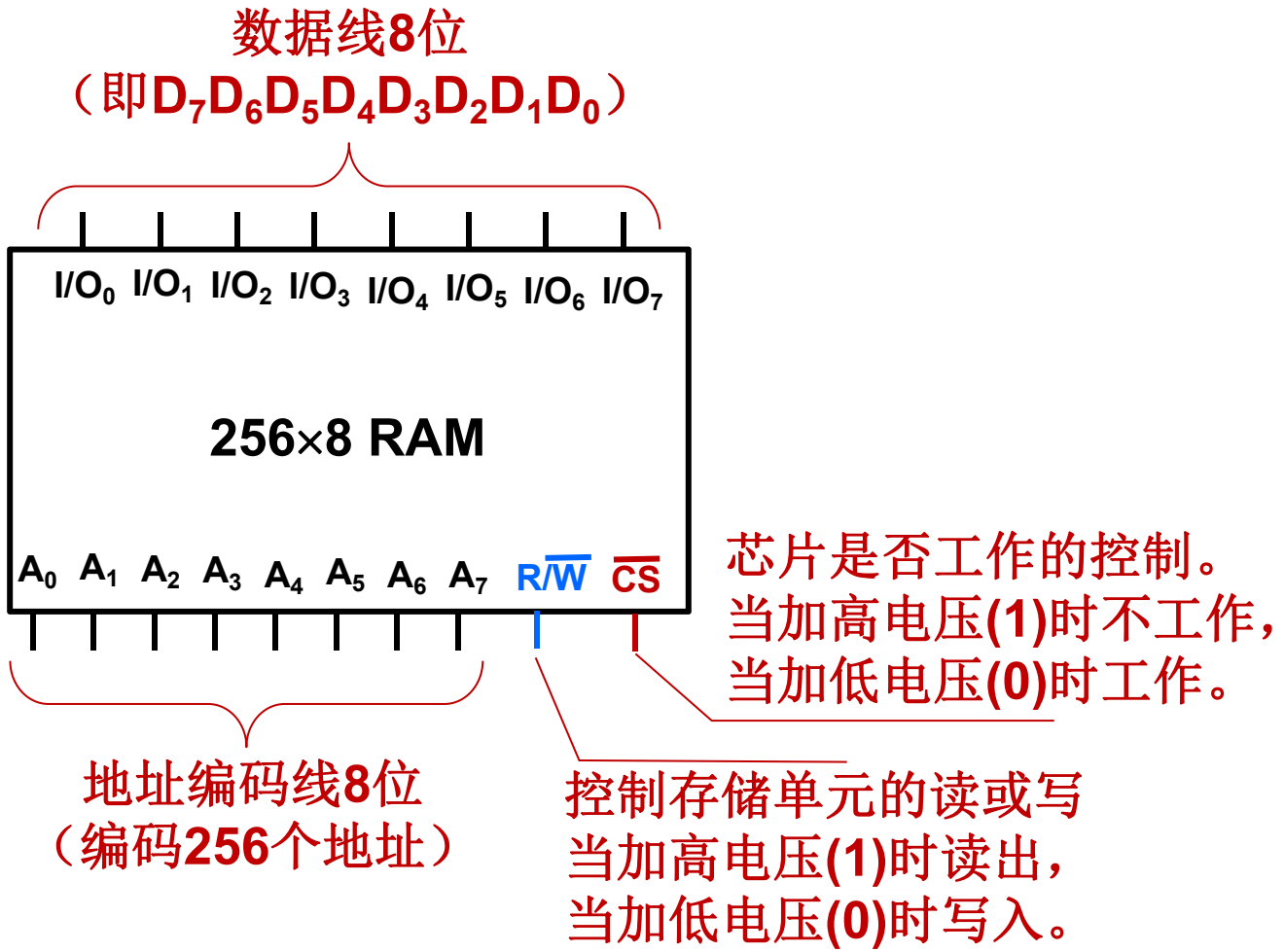
- 【或】阵列，表述了如何由$W_3W_2W_1W_0$的值产生D_3,D_2,D_1,D_0的值。单元控制线与数据线之间有黑点连接的，表示存储的是1，其能否输出取决于单元控制线是1还是0。由纵线向横线传输信号—下半部。
- 同一【数据线】上各连接点之间是 “或”关系。即各连接点只要有一个为1时，该数据线的信号为1，否则为0。

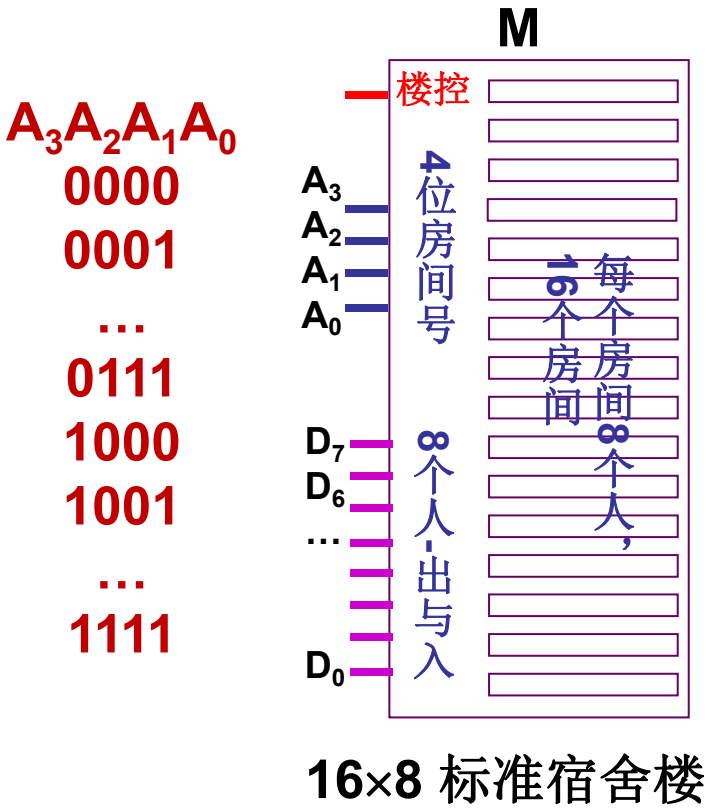
$$D_3 = W_0 \ OR \ W_2$$
$$D_2 = W_1 \ OR \ W_2 \ OR \ W_3$$
$$D_1 = W_1 \ OR \ W_2$$
$$D_0 = W_0 \ OR \ W_1 \ OR \ W_3$$



存储器容量不够怎么办?

存储器芯片 vs. 标准宿舍楼

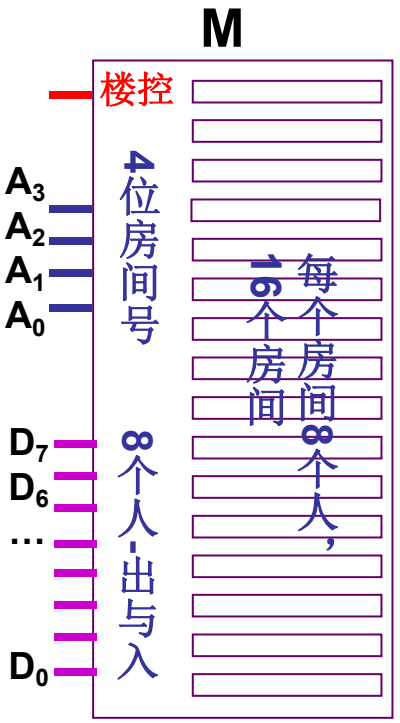




16×8 标准宿舍楼

存储器容量不够怎么办？

一个宿舍楼不够怎么办？



16×8 标准宿舍楼

一个标准宿舍楼：16个房间，每个房间住8人

$A_3A_2A_1A_0$　　$D_7…D_0$

住宿需求：64个大房间，每个大房间住16人？

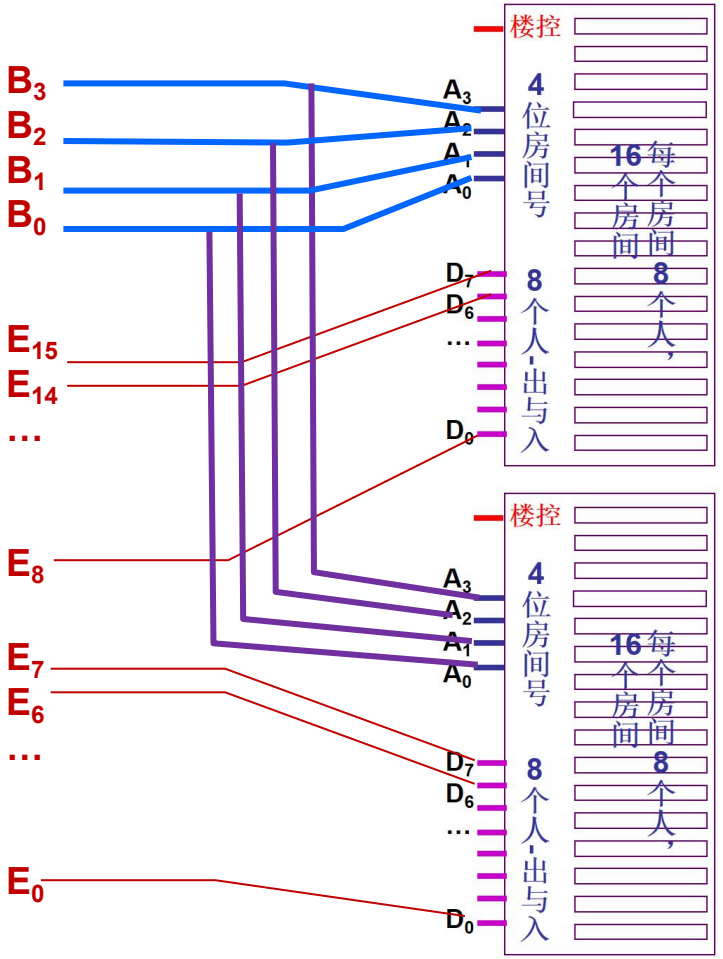
$B_5B_4\ B_3B_2B_1B_0$　　$E_{15}…E_8E_7…E_0$

存储器容量不够怎么办？

多个宿舍楼拼接在一起使用

住宿需求：
64个大房间，每个大房间住16人？
$B_5B_4B_3B_2B_1B_0$
$E_{15}…E_8E_7…E_0$

解决方案第1步：
2个标准宿舍楼对应相同地址$B_3B_2B_1B_0$的房间构成一个大房间，其中编号为$E_{15}…E_8$住上面的楼，而$E_7…E_0$住下面的楼，确保他们的地址$A_3A_2A_1A_0$在同一时刻是相同的，即同时进出。

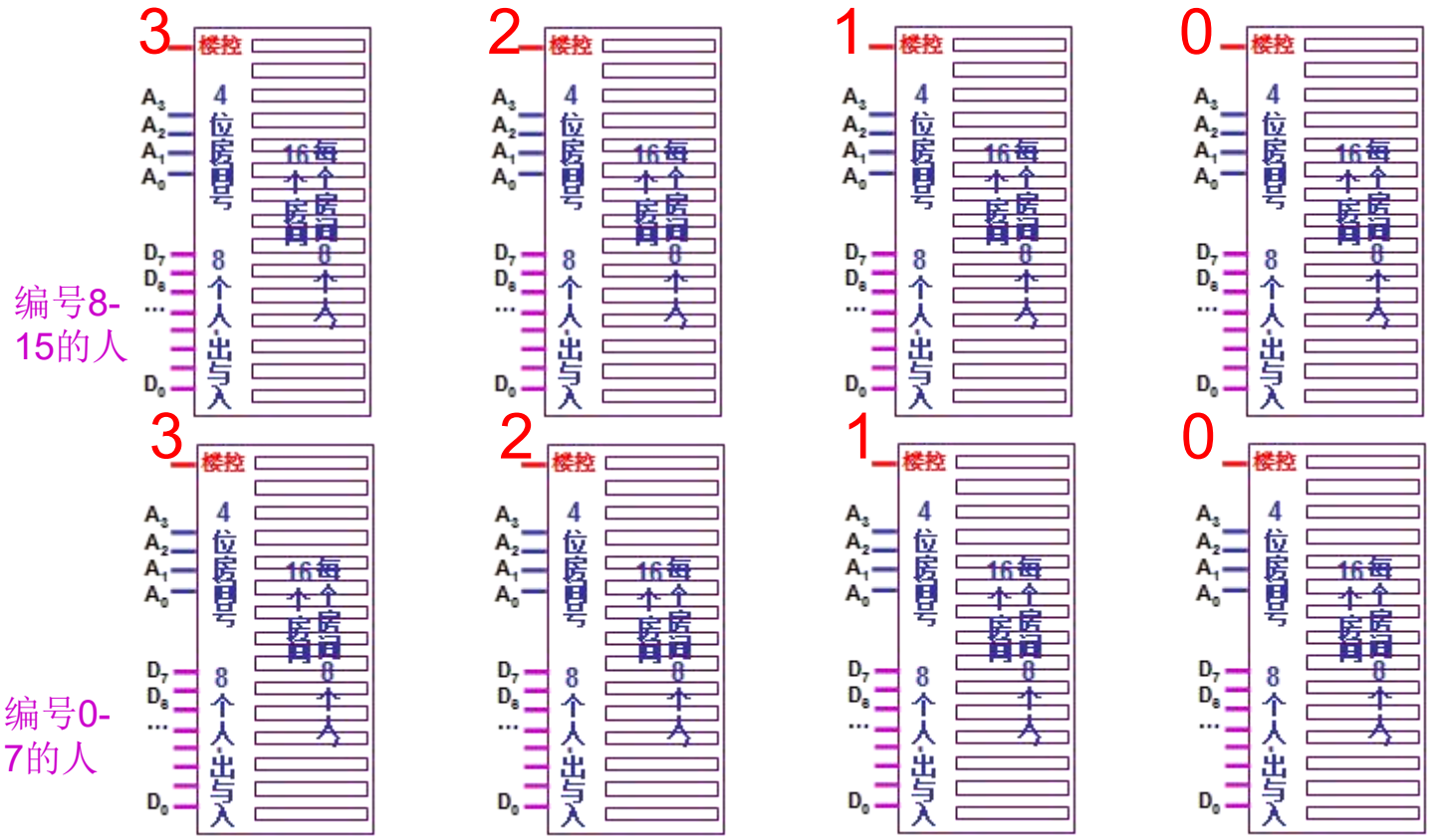


存储器容量不够怎么办?

多组宿舍楼联合使用

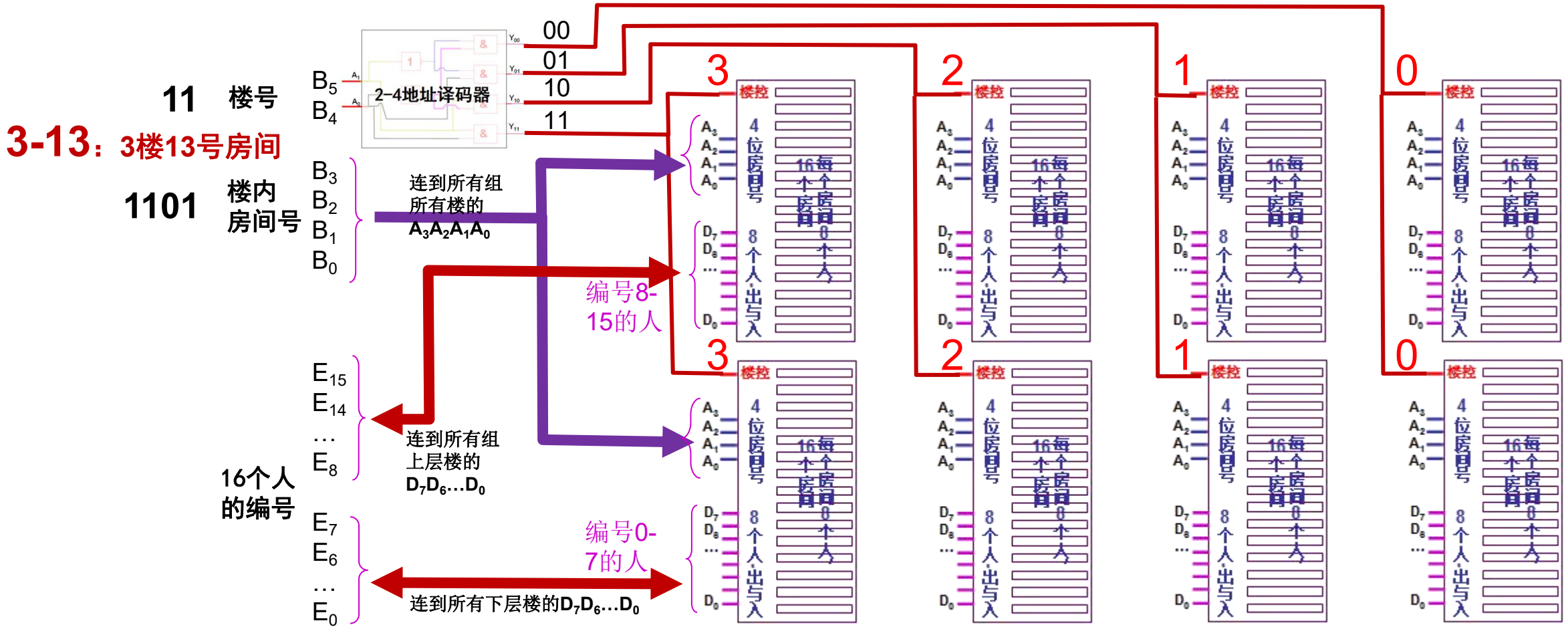
住宿需求:
64个大房间，每个大房间住16人?
$B_5B_4B_3B_2B_1B_0$
$E_{15}...E_8E_7...E_0$

解决方案第2步:
2个标准宿舍楼为一组，64个房间则需要4组（每组16个房间）。
用楼控信号控制哪一组。楼控信号为0,1,2,3，使用B_5B_4来产生楼控信号，使用$B_3B_2B_1B_0$来产生楼内的地址。



存储器容量不够怎么办?

用多个标准宿舍楼可组合出容量更大的宿舍楼



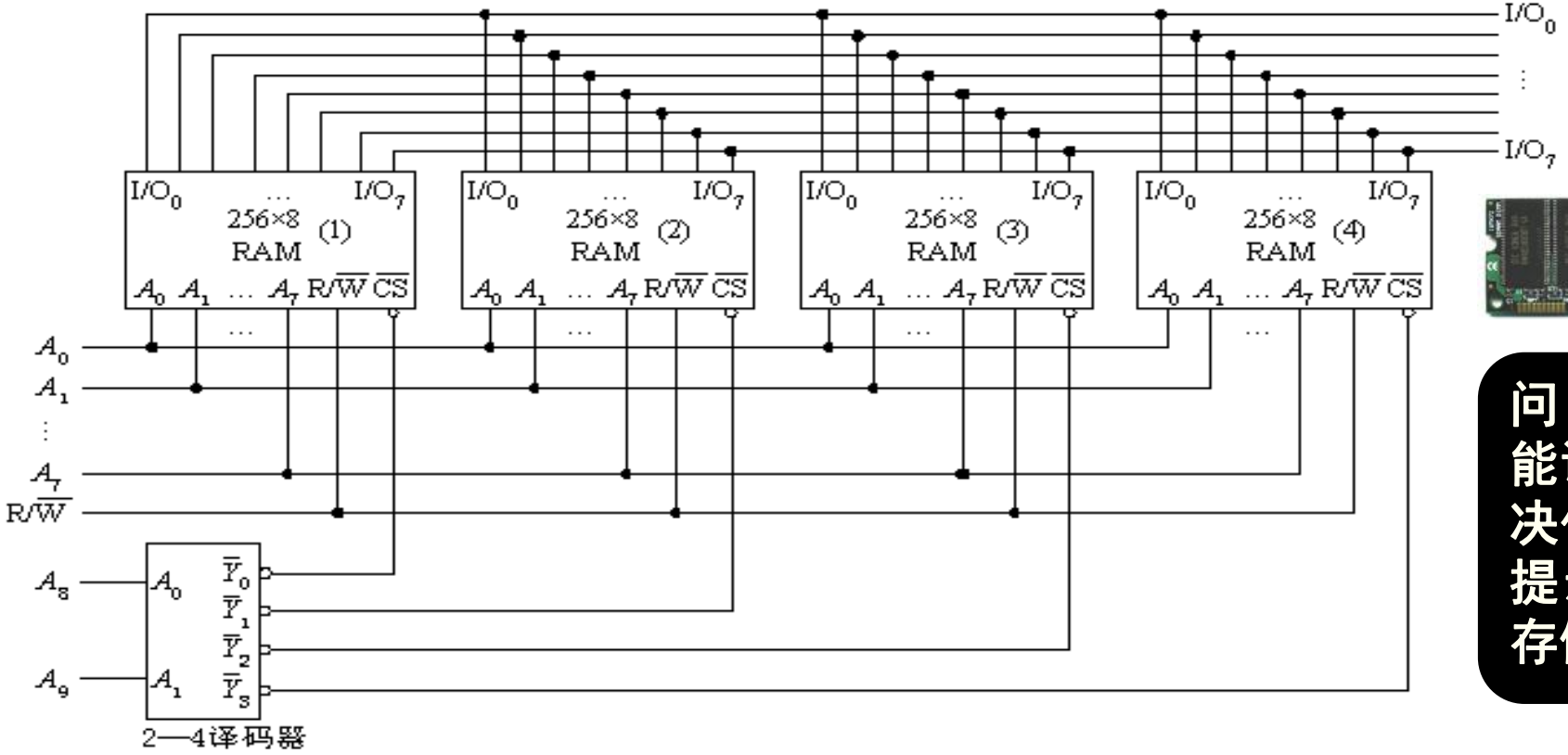
存储器容量不够怎么办?

用多个存储器芯片可搭建容量更大的存储器

利用4个256x8存储器芯片扩展出1024x8存储器的电路图

$A_7A_6\ldots A_0$ $D_7D_6\ldots D_0$

$A_9A_8A_7A_6\ldots A_0$ $D_7D_6\ldots D_0$



问：从概念的角度，你能说说存储器扩展要解决什么问题吗?
提示：地址编码空间，存储字长.

单选题 1分

设置

28

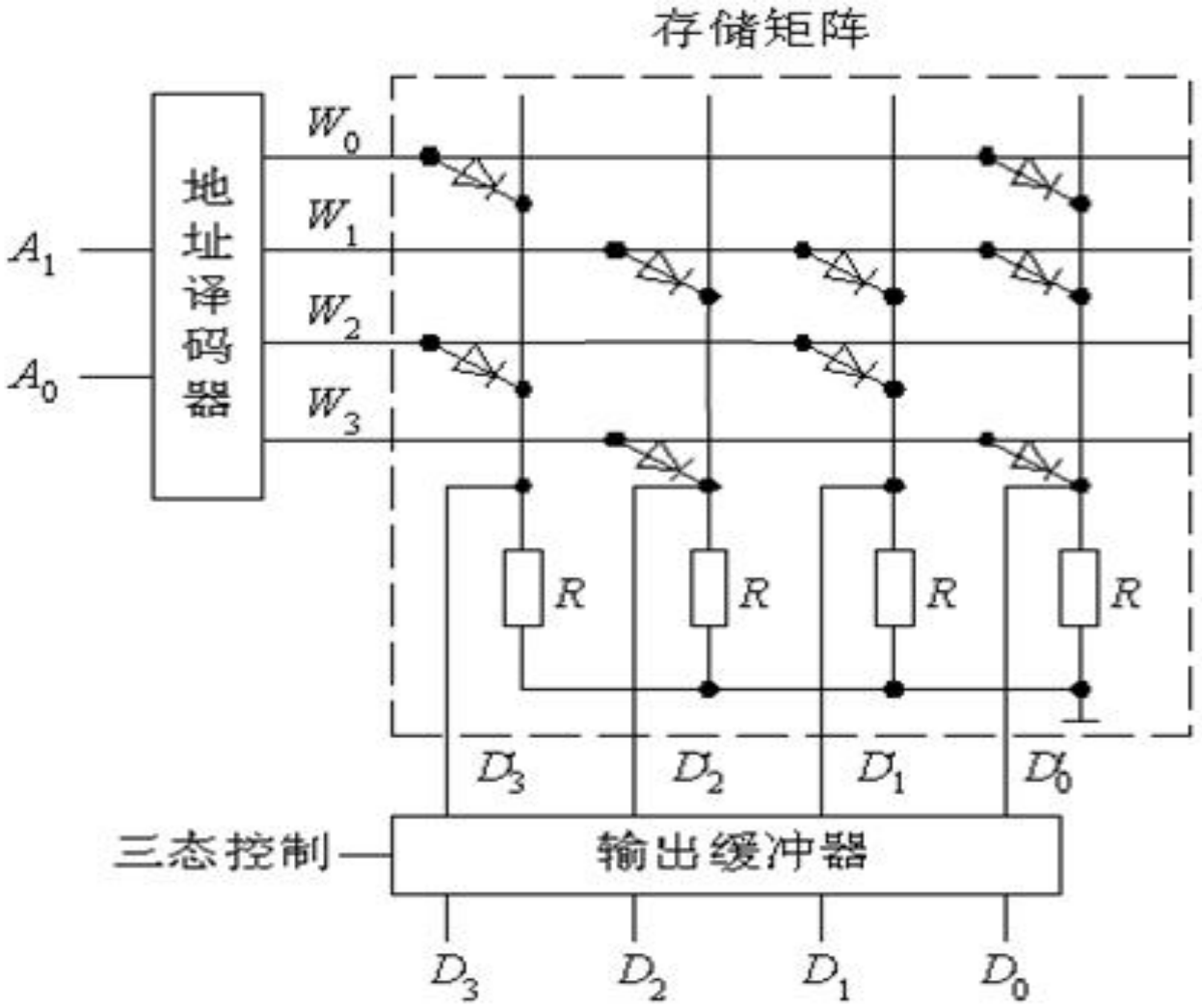
下列说法不正确的是_____。

A 该存储器可存取4个4位的存储单元；

B 该存储器其中的一个存储单元的内容是1010；

C 该存储器既可读出，又可写入；

D 该存储器的地址码分别是00，01，10和11。



提交

机器指令与机器程序

计算机如何计算一个运算式?

$$8\times3^2+2\times3+6$$

机器指令与机器程序

机器级算法

算法---机器可以执行的求解问题的规则及步骤。

计算 $8\times3^2+2\times3+6 = ((8\times3)+2)\times3+6$

计算方法1

Step1: 取出数3至运算器中
Step2: 乘以数3在运算器中
Step3: 乘以数8在运算器中
Step4: 存结果8×3^2在存储器中
Step5: 取出数2至运算器中
Step6: 乘以数3在运算器中
Step7: 加上(8×3^2)在运算器中
Step8: 加上数6在运算器中

计算方法2

Step1: 取出数3至运算器中
Step2: 乘以数8在运算器中
Step3: 加上数2在运算器中
Step4: 乘以数3在运算器中
Step5: 加上数6至运算器中

问：怎么看待算法节省的步数？---算法需要“优化”

机器指令与机器程序

机器指令

机器指令--是CPU可以直接分析并执行的指令，一般由0和1的编码表示。

指令 ≈ 操作码 + 地址码；

操作码	地址码
000001	00 00000111
(如取数，加法等操作)	(操作中的数据的来源)

000100 0000001010

000100 0000000100

000011 0000001100

000011 0000001000

机器指令		对应的功能
操作码	地址码	
取数	α	α号存储单元的数 取出送到运算器；
000001	0000000100	
存数	β	运算器中的数 存储到 β号存储单元；
000010	0000010000	
加法	γ	运算器中的数 加上 γ号存储单元的数，结果保留在运算器；
000011	0000001010	
乘法	δ	运算器中的数 乘以 δ号存储单元的数，结果保留在运算器；
000100	0000001001	
打印		打印指令
000101	0000001100	
停机		停机指令
000110		

机器语言—机器能够执行的所有指令的集合。

机器指令与机器程序

机器程序：将机器级算法用机器指令进行表达

假设数字3、8、2、6分别存储在8号、9号、10号和11号存储单元

$((8\times3)+2)\times3+6$

计算方法2

Step1: 取出数3至运算器中

Step2: 乘以数8在运算器中

Step3: 加上数2在运算器中

Step4: 乘以数3在运算器中

Step5: 加上数6至运算器中

机器级算法

```
000001  0000001000
000100  0000001001
000011  0000001010
000100  0000001000
000011  0000001011
000010  0000001100
000101  0000001100
000110  0000000000
```

机器程序

机器指令		对应的功能
操作码	地址码	
取数	α	α号存储单元的数 取出送到运算器；
000001	0000000100	
存数	β	运算器中的数 存储到 β号存储单元；
000010	0000010000	
加法	γ	运算器中的数 加上 γ号存储单元的数，结果保留在运算器；
000011	0000001010	
乘法	δ	运算器中的数 乘以 δ号存储单元的数，结果保留在运算器；
000100	0000001001	
打印		打印指令
000101	0000001100	
停机		
000110		

机器语言与机器指令

机器指令与机器程序

将机器程序和数据装载进存储器中?

计算8　3^2+2　3+6的程序

```
000001  0000001000
000100  0000001001
000011  0000001010
000100  0000001000
000011  0000001011
000010  0000001100
000101  0000001100
000110  0000000000
```

机器程序

地址（编号）	存储单元
00000000 00000000	0000010000001000
00000000 00000001	0001000000001001
00000000 00000010	0000110000001010
00000000 00000011	0001000000001000
00000000 00000100	0000110000001011
00000000 00000101	0000100000001100
00000000 00000110	0001010000001100
00000000 00000111	0001100000000000
00000000 00001000	0000000000000011
00000000 00001001	0000000000001000
00000000 00001010	0000000000000010
00000000 00001011	0000000000000110
00000000 00001100	

存储器

程序

数据

机器指令与机器程序

改改机器程序，体验机器程序

计算8　3^2+2　3+6的程序

计算5　4^2+3　4+7的程序

计算ax^2+bx+c的程序？

000001 0000001000
000100 0000001001
000011 0000001010
000100 0000001000
000011 0000001011
000010 0000001100
000101 0000001100
000110 0000000000

机器程序

地址（编号）	存储单元
00000000 00000000	0000010000001000
00000000 00000001	0001000000001001
00000000 00000010	0000110000001010
00000000 00000011	0001000000001000
00000000 00000100	0000110000001011
00000000 00000101	0000100000001100
00000000 00000110	0001010000001100
00000000 00000111	0001100000000000
00000000 00001000	0000000000000011
00000000 00001001	0000000000001000
00000000 00001010	0000000000000010
00000000 00001011	0000000000000110
00000000 00001100	

存储器

程序

数据

机器指令与机器程序

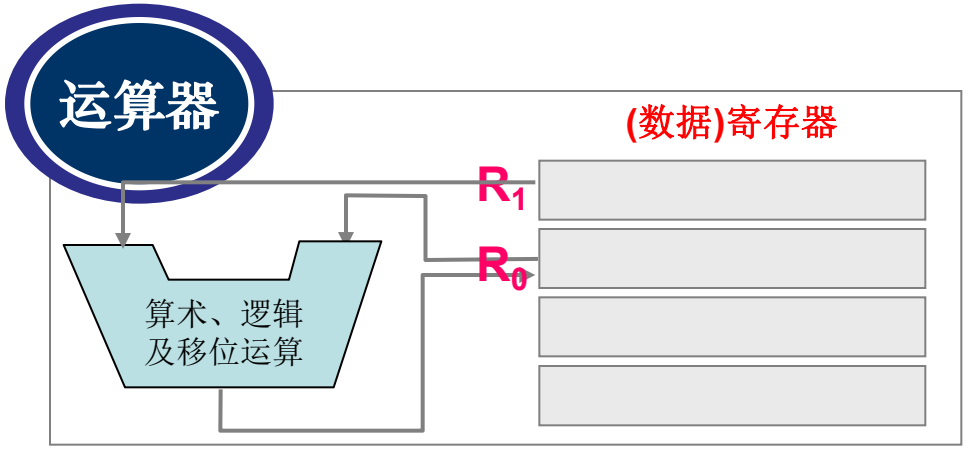
改改机器程序，体验机器程序

对应的十进制地址	存储单元的地址	存储单元的内容		说明
		操作码	地址码	
0	00000000 00000000	000001	0000001000	指令：取出 8 号存储单元的数(即 3)至运算器中
1	00000000 00000001	000100	0000001001	指令：乘以 9 号存储单元的数(即 8)得 8×3 在运算器中
2	00000000 00000010	000011	0000001010	指令：加上 10 号存储单元的数(即 2)得 8×3+2 在运算器中
3	00000000 00000011	000100	0000001000	指令：乘以 8 号存储单元的数(即 3) 得(8×3+2)×3 在运算器中
4	00000000 00000100	000011	0000001011	指令：加上 11 号存储单元的数(即 6)得 $8\times3^2+2\times3+6$ 至运算器中
5	00000000 00000101	000010	0000001100	指令：将上述运算器中结果存于 12 号存储单元
6	00000000 00000110	000101	0000001100	指令：打印
7	00000000 00000111	000110		指令：停机
8	00000000 00001000	000000 0000000011		数据：数 3 存于 8 号单元
9	00000000 00001001	000000 0000001000		数据：数 8 存于 9 号单元
10	00000000 00001010	000000 0000000010		数据：数 2 存于 10 号单元
11	00000000 00001011	000000 0000000110		数据：数 6 存于 11 号单元
12	00000000 00001100			数据：存放结果

程序与数据以同等地位存于存储器中

一台典型的计算机

运算器

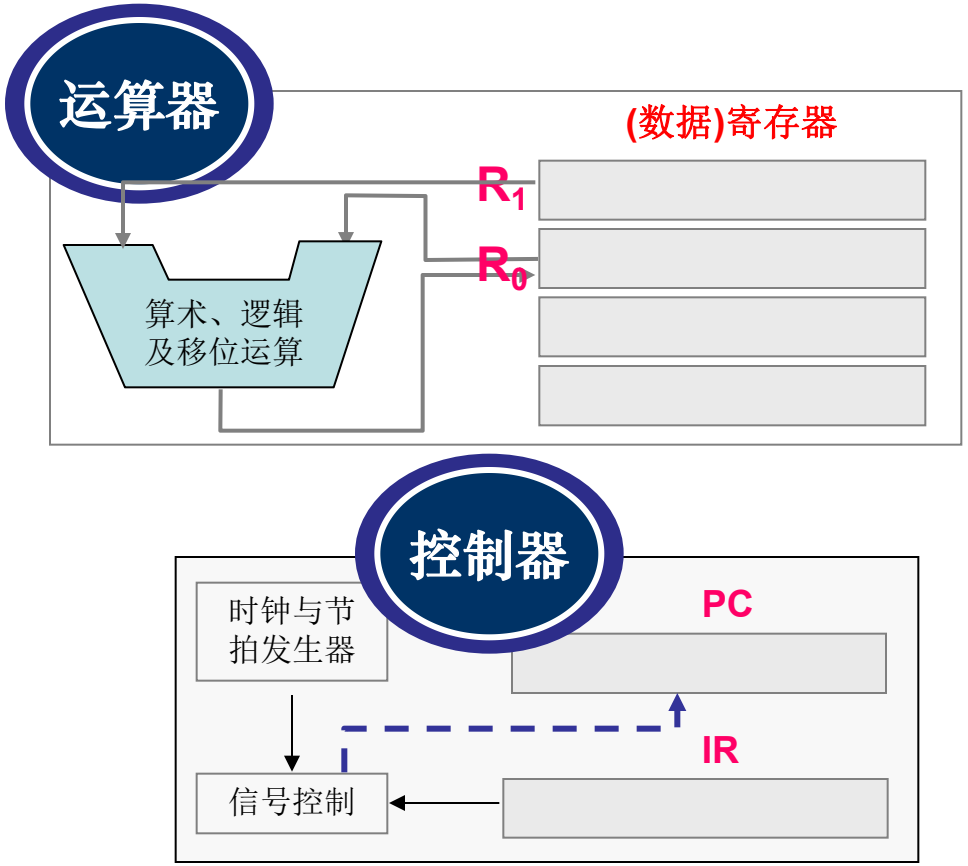


- (数据)寄存器
- 算术逻辑部件

$R_0 = R_1\ \theta\ R_0$

(赋值, R_0既是一个操作数，又保存运算结果)。
其中θ为算术、逻辑及移位运算符

一台典型的计算机



控制器

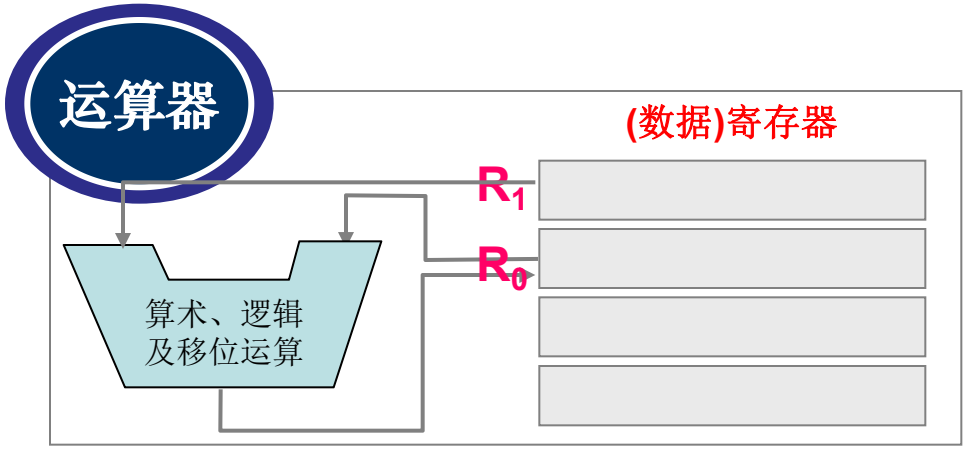
- 程序计数器PC
- 指令寄存器
- 信号控制器
- 时钟与节拍发生器

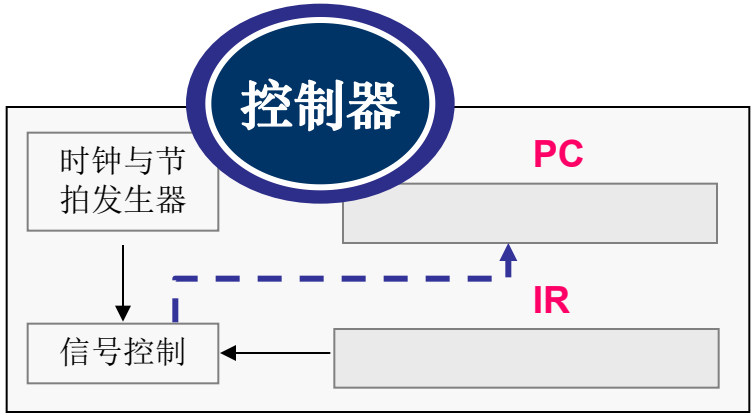
注：

PC：程序计数器---存储下一要执行指令的地址

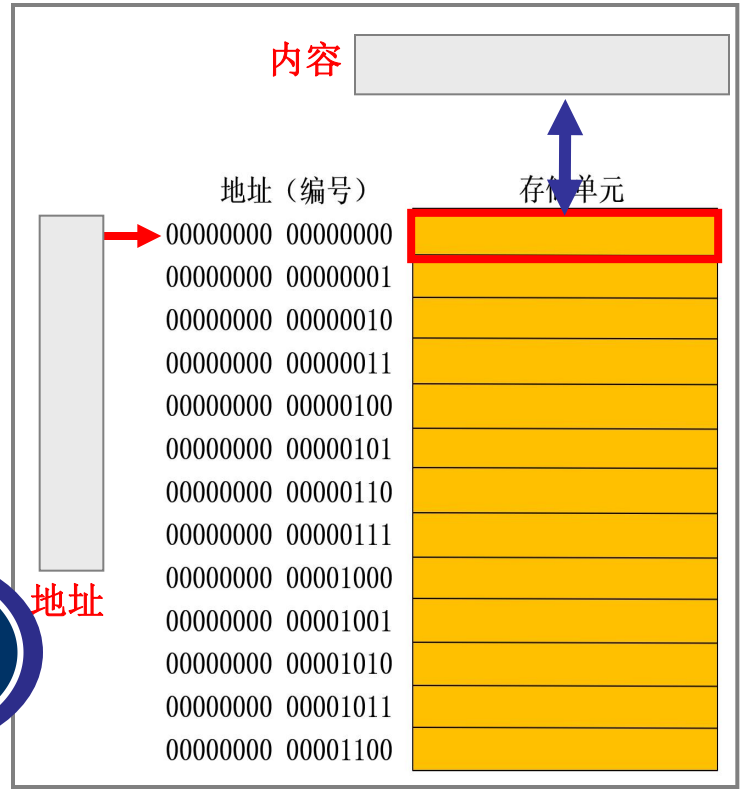
IR：指令寄存器---存储当前指令内容

一台典型的计算机





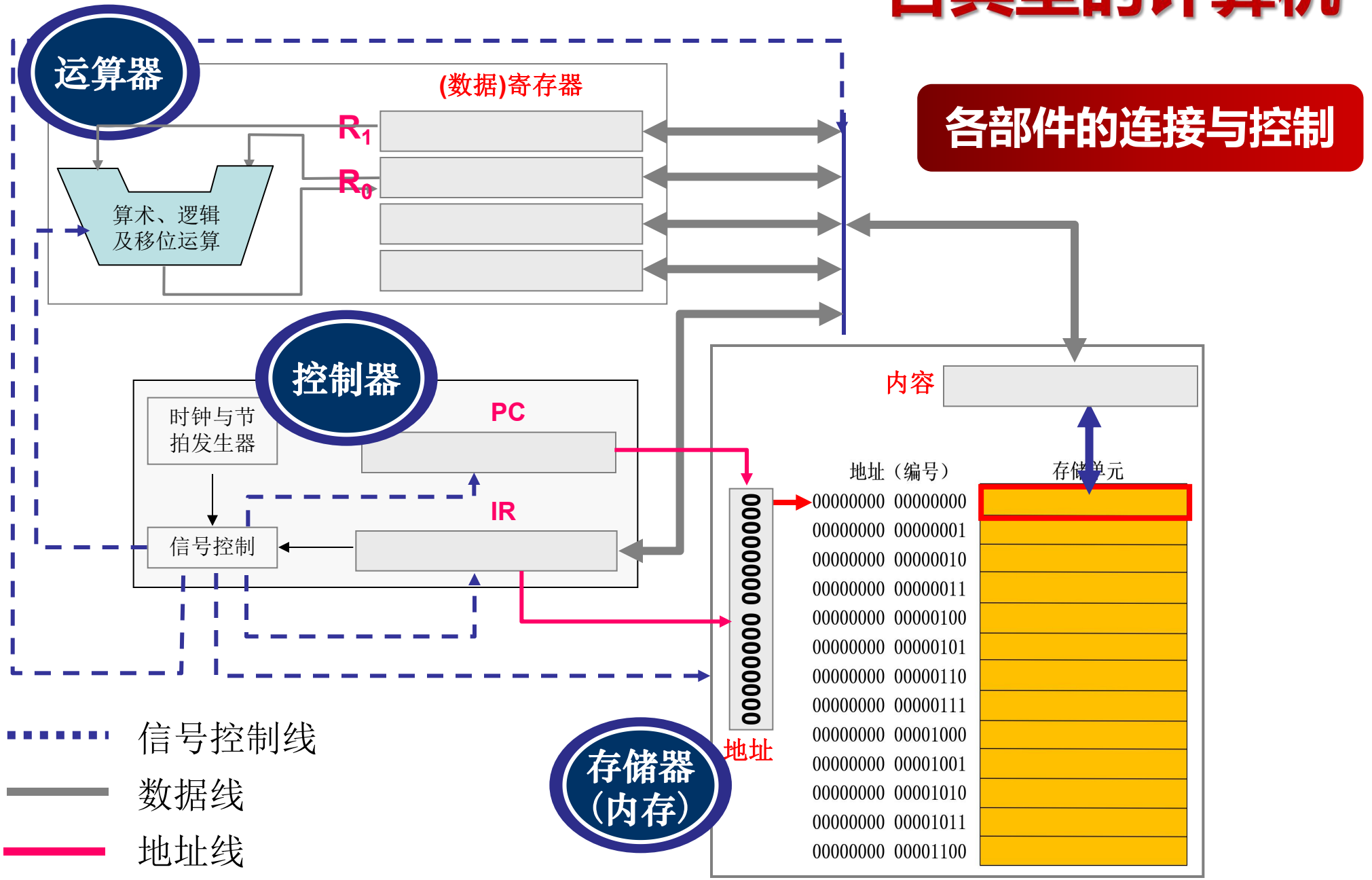
存储器



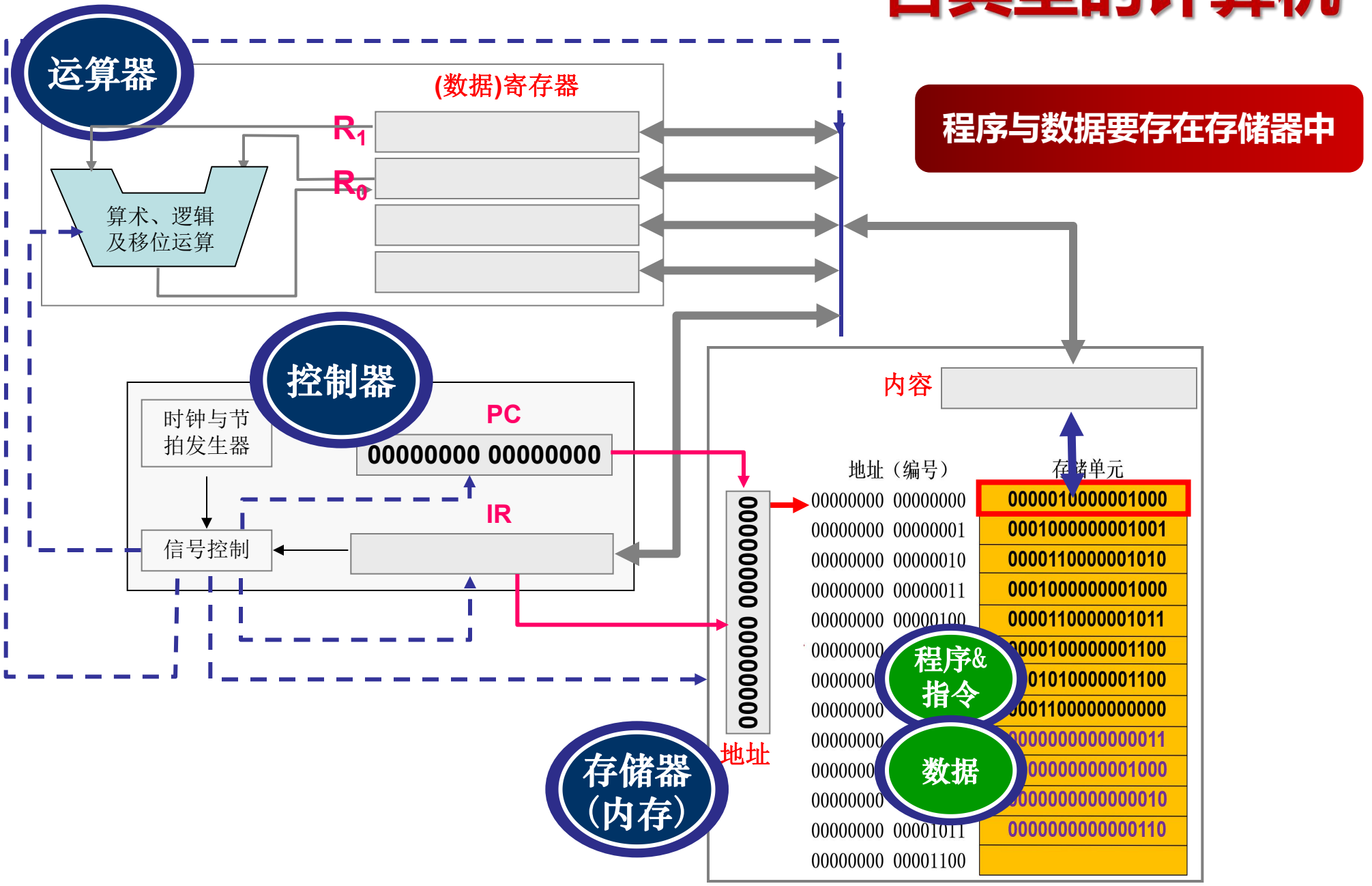
- 存储单元地址
- 存储单元内容

存储器（内存）

一台典型的计算机


运算器
(数据)寄存器
R1
R0
算术、逻辑
及移位运算
各部件的连接与控制
控制器
时钟与节
拍发生器
PC
IR
信号控制
内容
地址（编号）
存储单元
00000000 00000000
00000000 00000001
00000000 00000010
00000000 00000011
00000000 00000100
00000000 00000101
00000000 00000110
00000000 00000111
00000000 00001000
00000000 00001001
00000000 00001010
00000000 00001011
00000000 00001100
00000000 00000000
地址
存储器
(内存)
信号控制线
数据线
地址线

一台典型的计算机


运算器
(数据)寄存器
R1
R0
算术、逻辑
及移位运算
程序与数据要存在存储器中
控制器
时钟与节
拍发生器
PC
00000000 00000000
IR
信号控制
内容
地址（编号）
存储单元
00000000 00000000
00000000 00000001
00000000 00000010
00000000 00000011
00000000 00000100
00000000 00000101
00000000 00000110
00000000 00000111
00000000 00001000
00000000 00001001
00000000 00001010
00000000 00001011
00000000 00001100
0000010000001000
0001000000001001
0000110000001010
0001000000001000
0000110000001011
0000100000001100
0001010000001100
0001100000000000
0000000000000011
0000000000001000
0000000000000010
0000000000000110
00000000 00000000
地址
存储器
（内存）
程序&
指令
数据

一台典型的计算机

计算机各部件内部的简单构成关系

运算器

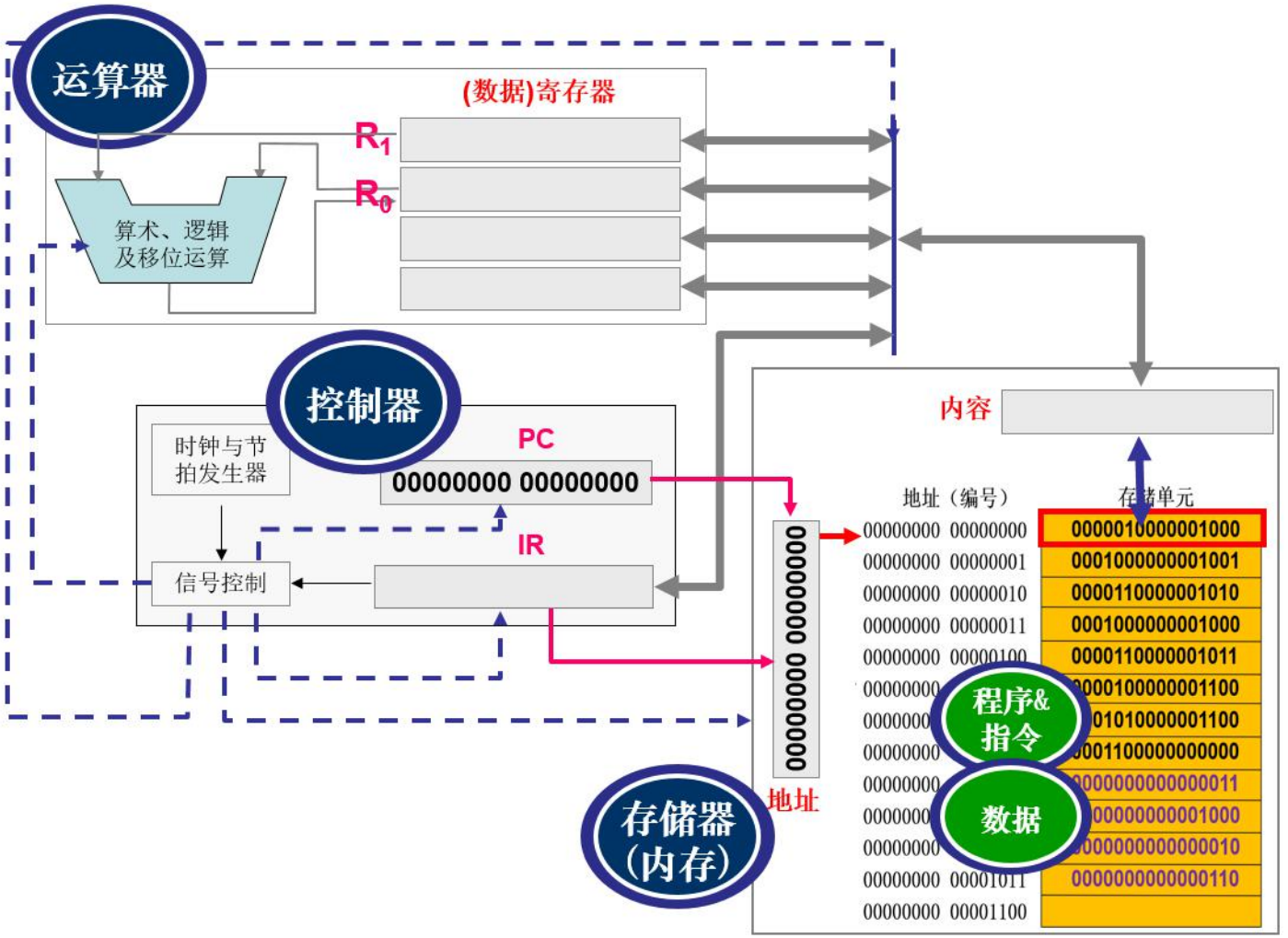
- 寄存器
- 算术逻辑部件

控制器

- 程序计数器PC
- 指令寄存器
- 信号控制器
- 时钟与节拍发生器

存储器

- 存储单元地址
- 存储单元内容



取数指令语义动作：00000001 00001000

1. 取指令：发送指令地址给存储器
   PC中的地址发给存储器，
   信号发生器通知存储器工作
2. 取指令：取出存储器指令给IR
   存储器对地址进行译码：找到对应存储单元，将数据读到内容寄存器；
   控制器令IR接收内容寄存器中的指令
   操作码传递给信号发生器
3. 执行并分析指令：
   PC内容加1
   取出指令中的地址码，发给存储器
   信号发生器通知存储器工作
4. 执行并分析指令：
   存储器对地址进行译码：找到对应存储单元，将数据读到内容寄存器；
   信号发生器通知R0寄存器接收数据

指令执行

◆不同的指令，由一组不同的电信号构成。有些电信号需要按次序完成。

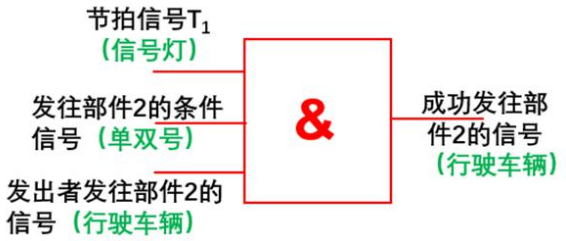
◆最小的时间区隔单位--时钟周期。不同的时钟周期状态称为节拍。

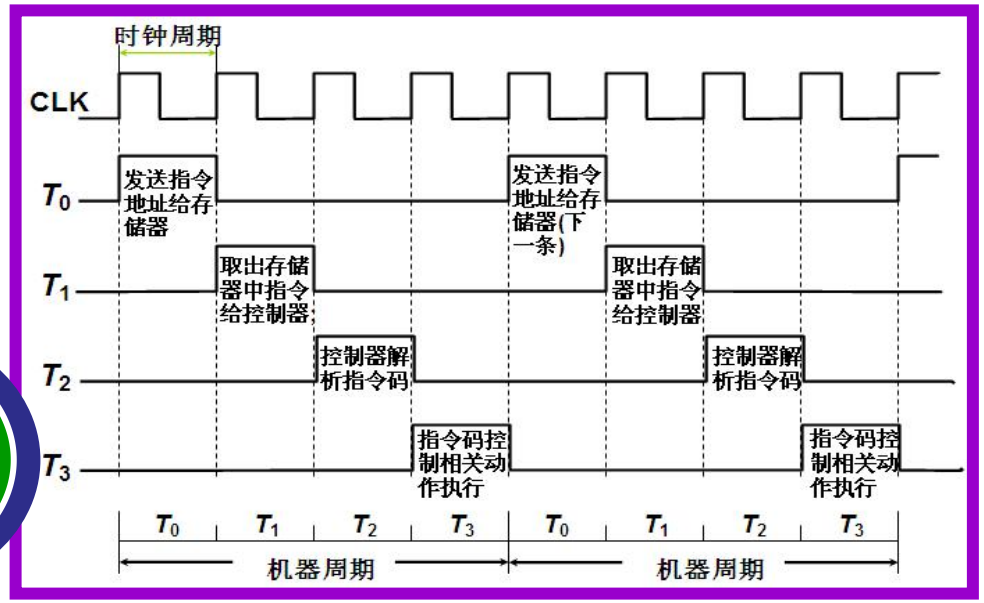
◆多个节拍构成一个机器周期。一条指令占用一个或多个机器周期。

◆同一指令的电信号在时钟与节拍的控制下按次序产生与传输。

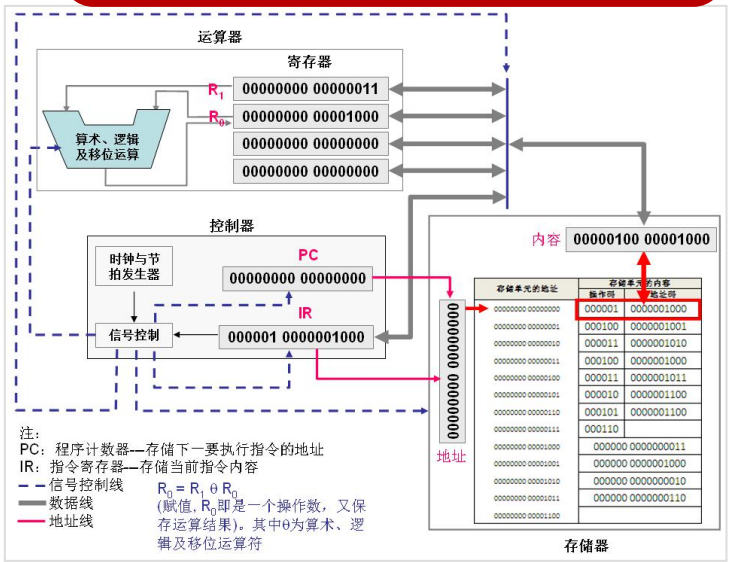
指令执行

指令执行的信号化--即在节拍控制下有序地发出各种电信号





时钟周期、节拍与机器周期



问：机器的“主频”指的是什么？

多选题 1分

设置

44

关于“存储在存储器中程序的执行”问题，下列说法正确的是

A 机器需要提供一个其可以执行的指令集合；

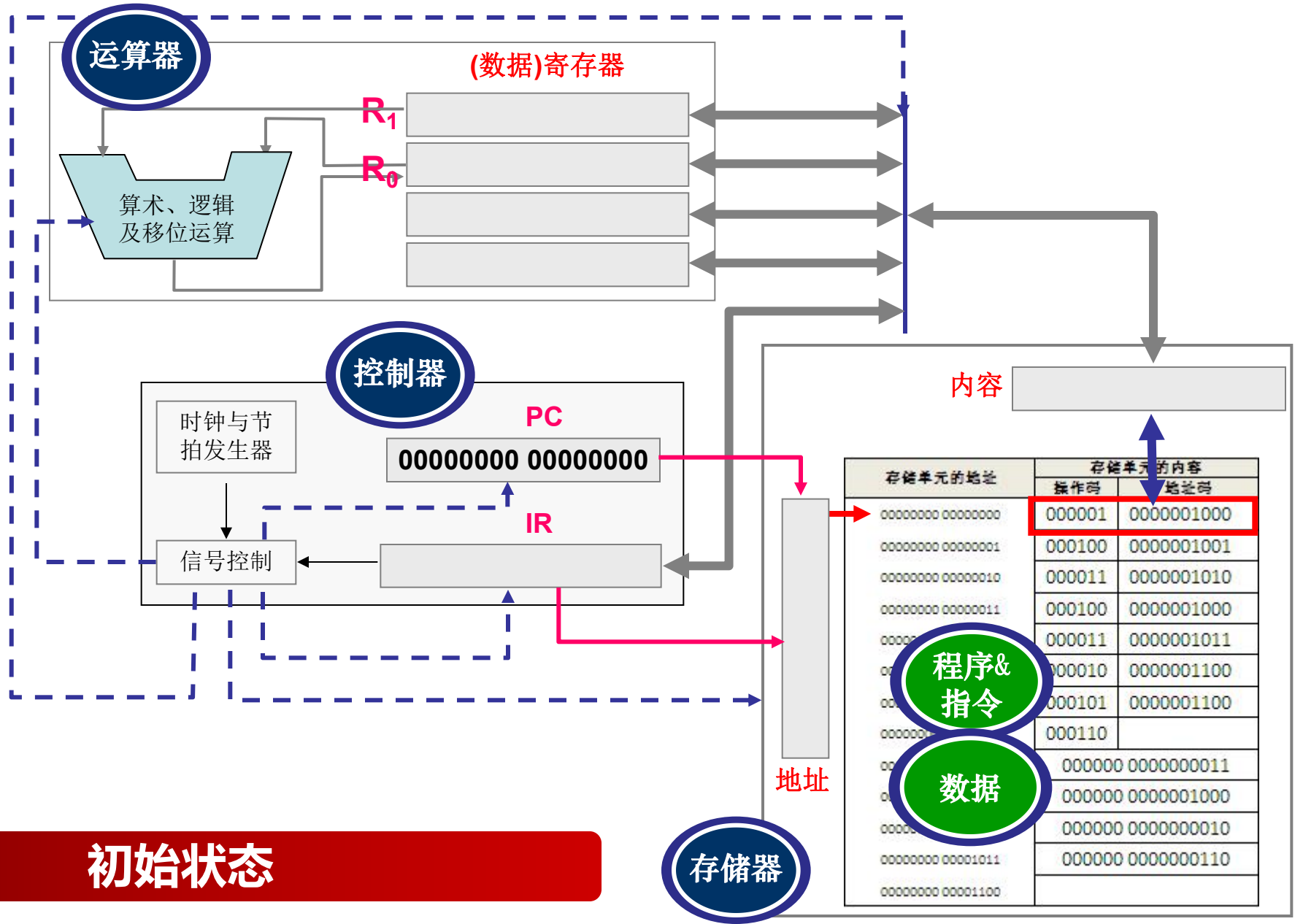
B 人们用指令集合中的指令编写程序，并将编写好的程序和数据事先存放于存储器中；

C 控制器一条接一条的从存储器中读取指令，读取一条指令则执行一条指令，一条指令执行完成后，再读下一条指令；

D 当读取一条指令后，程序计数器PC的值自动加1，以指向下一条将要读取的指令；当程序需要转往它处执行时，则可以它处存放指令的地址来修改PC的值即可；

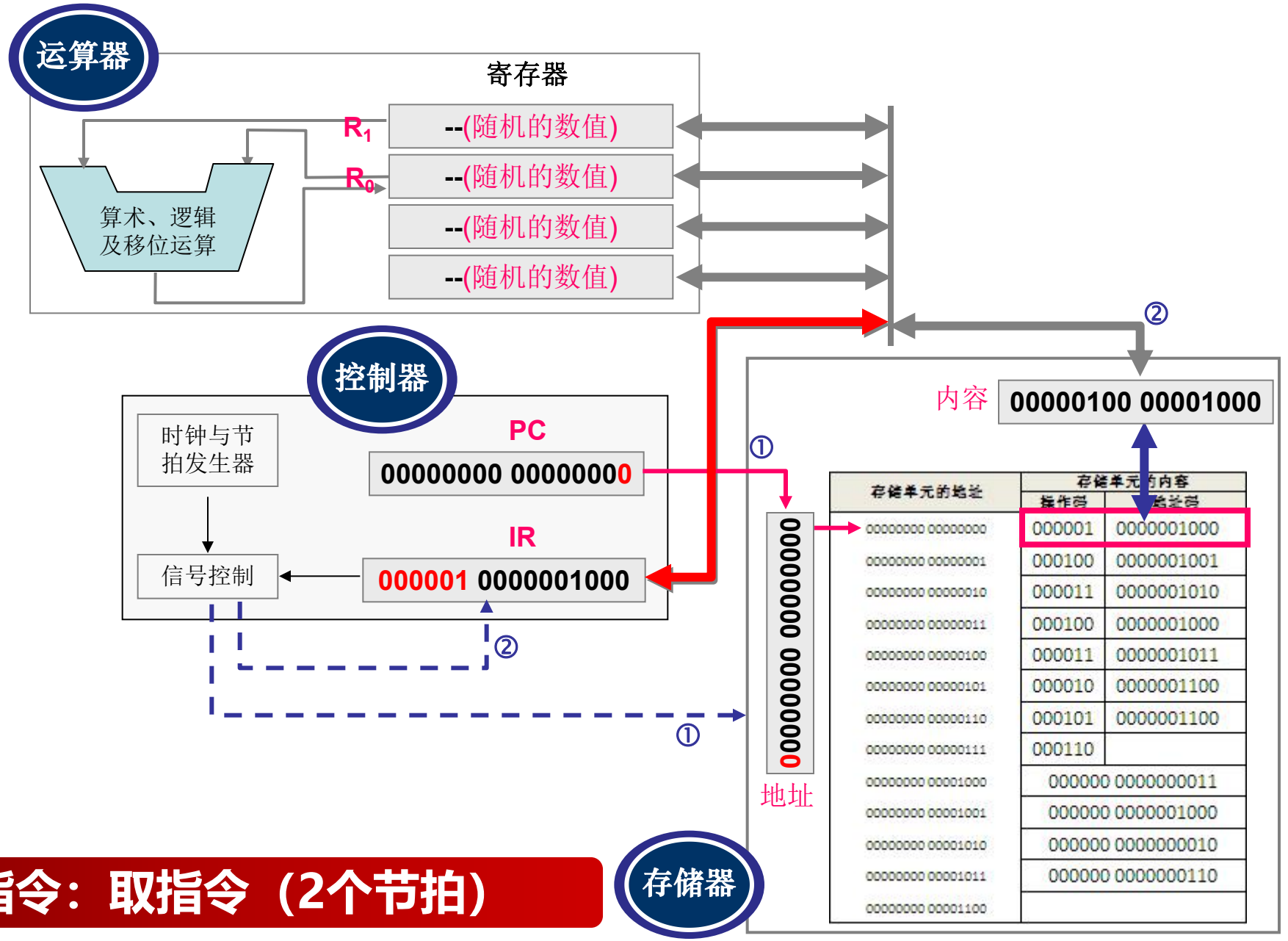
提交

机器程序的执行过程模拟



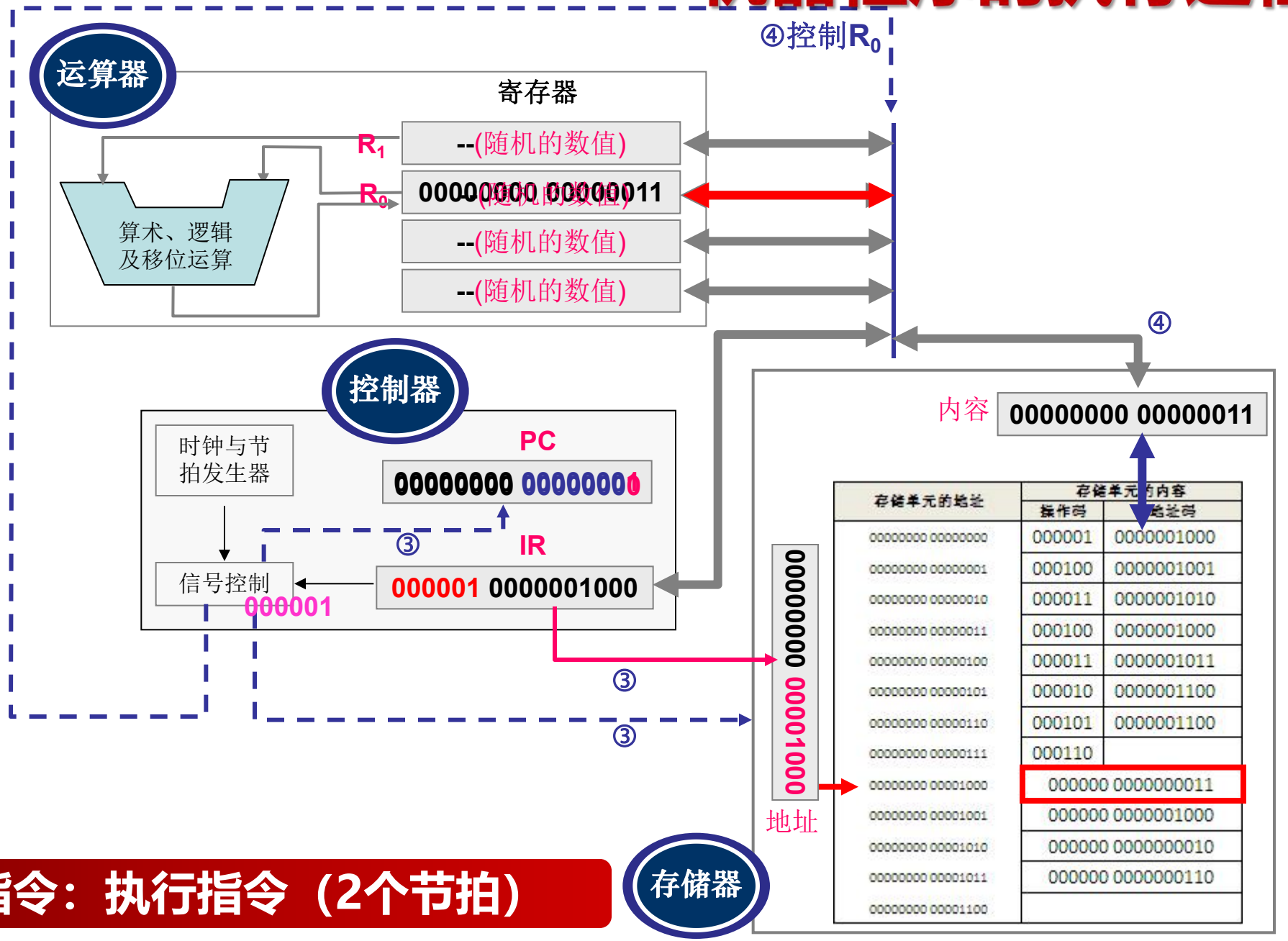
初始状态

机器程序的执行过程模拟



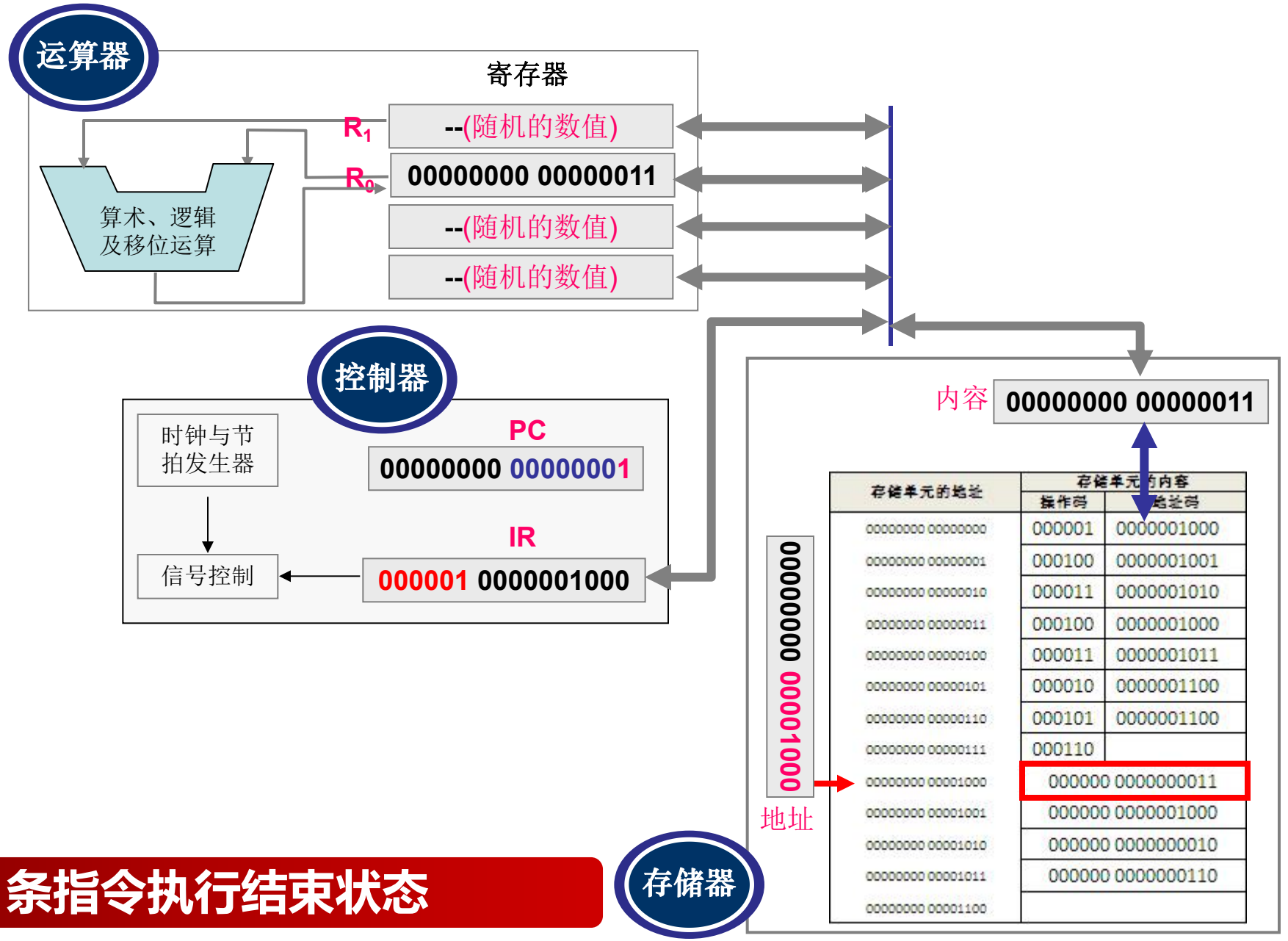
第1条指令：取指令（2个节拍）

机器程序的执行过程模拟



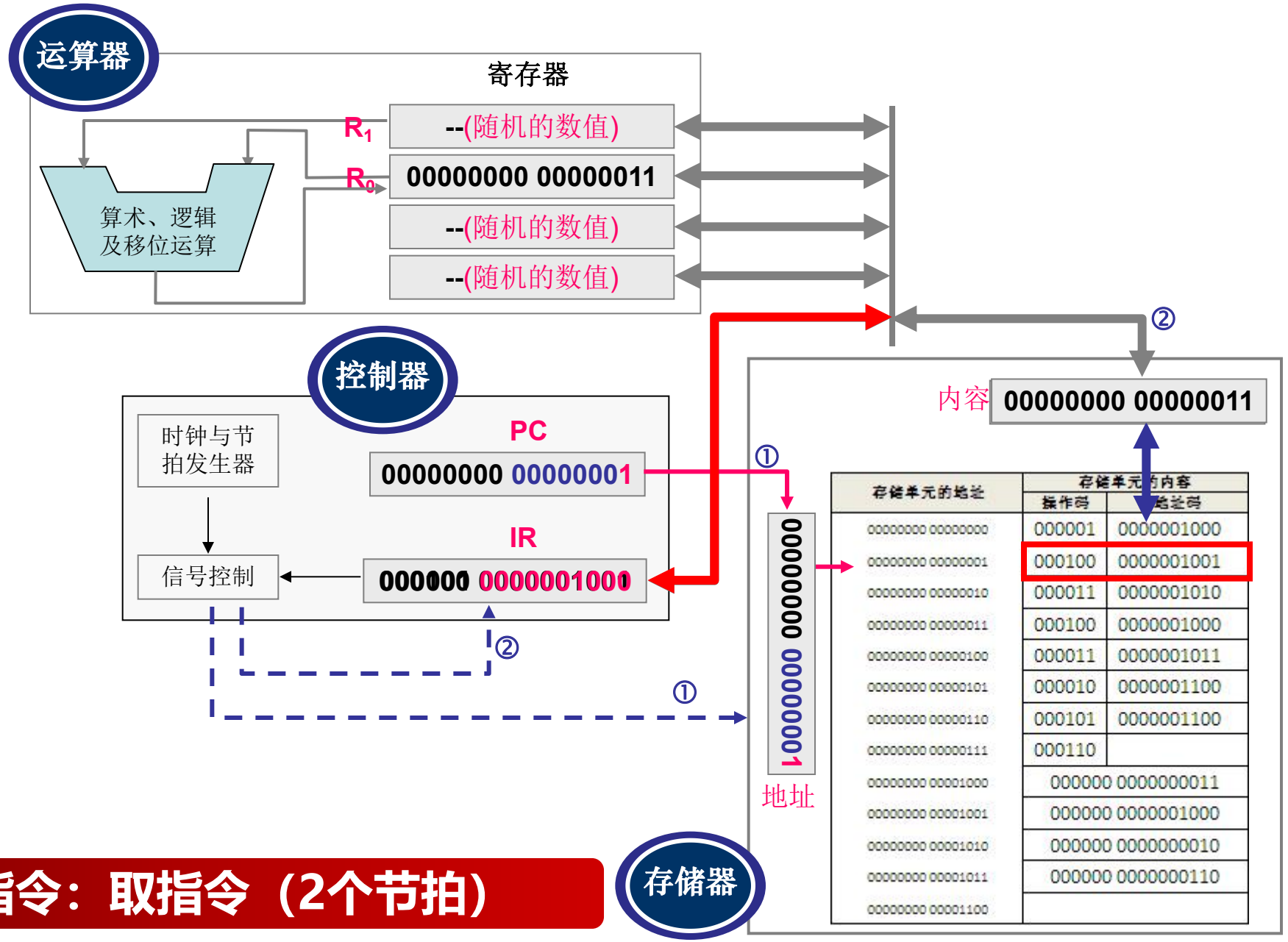
第1条指令：执行指令（2个节拍）

机器程序的执行过程模拟



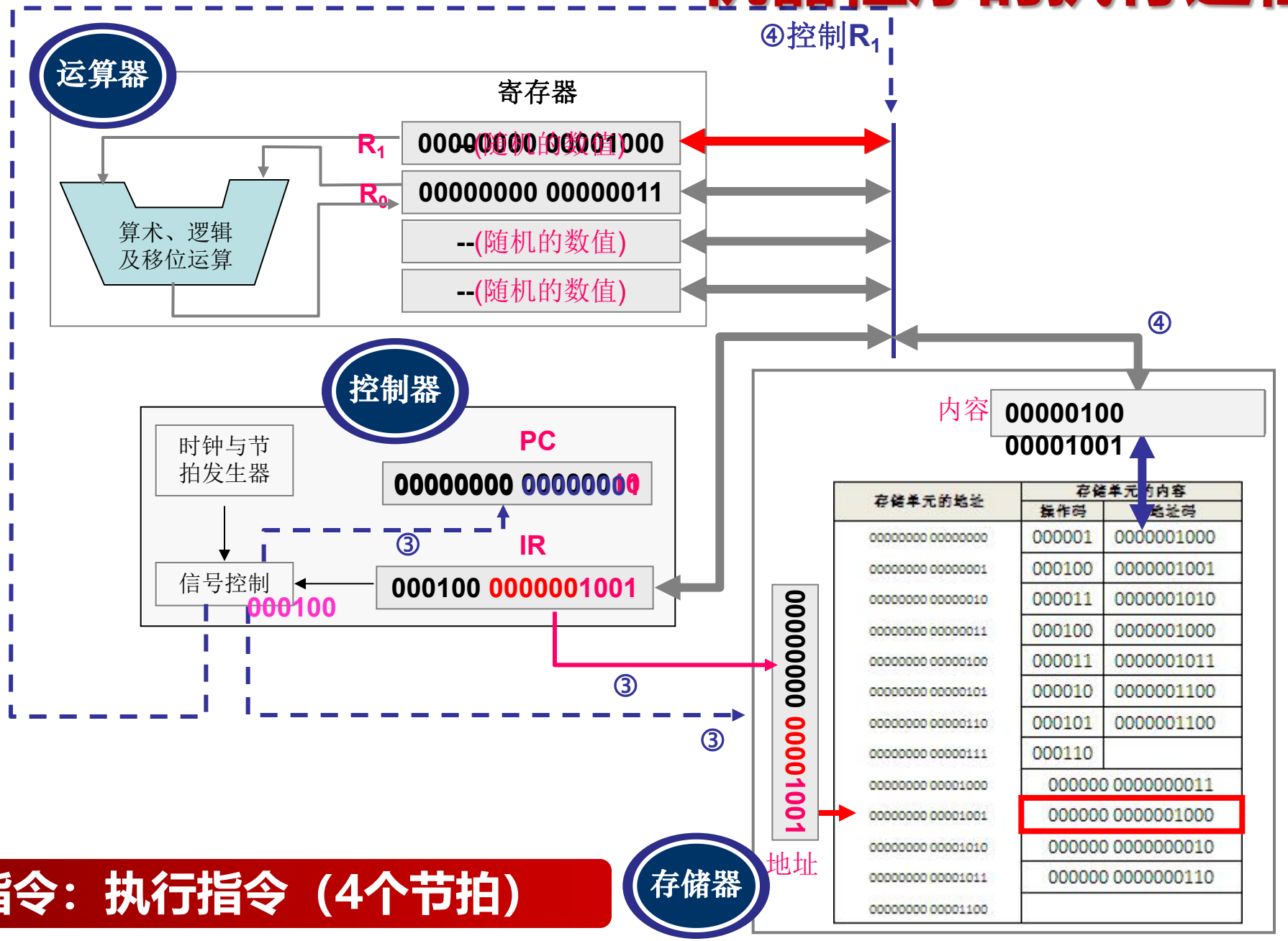
第1条指令执行结束状态

机器程序的执行过程模拟



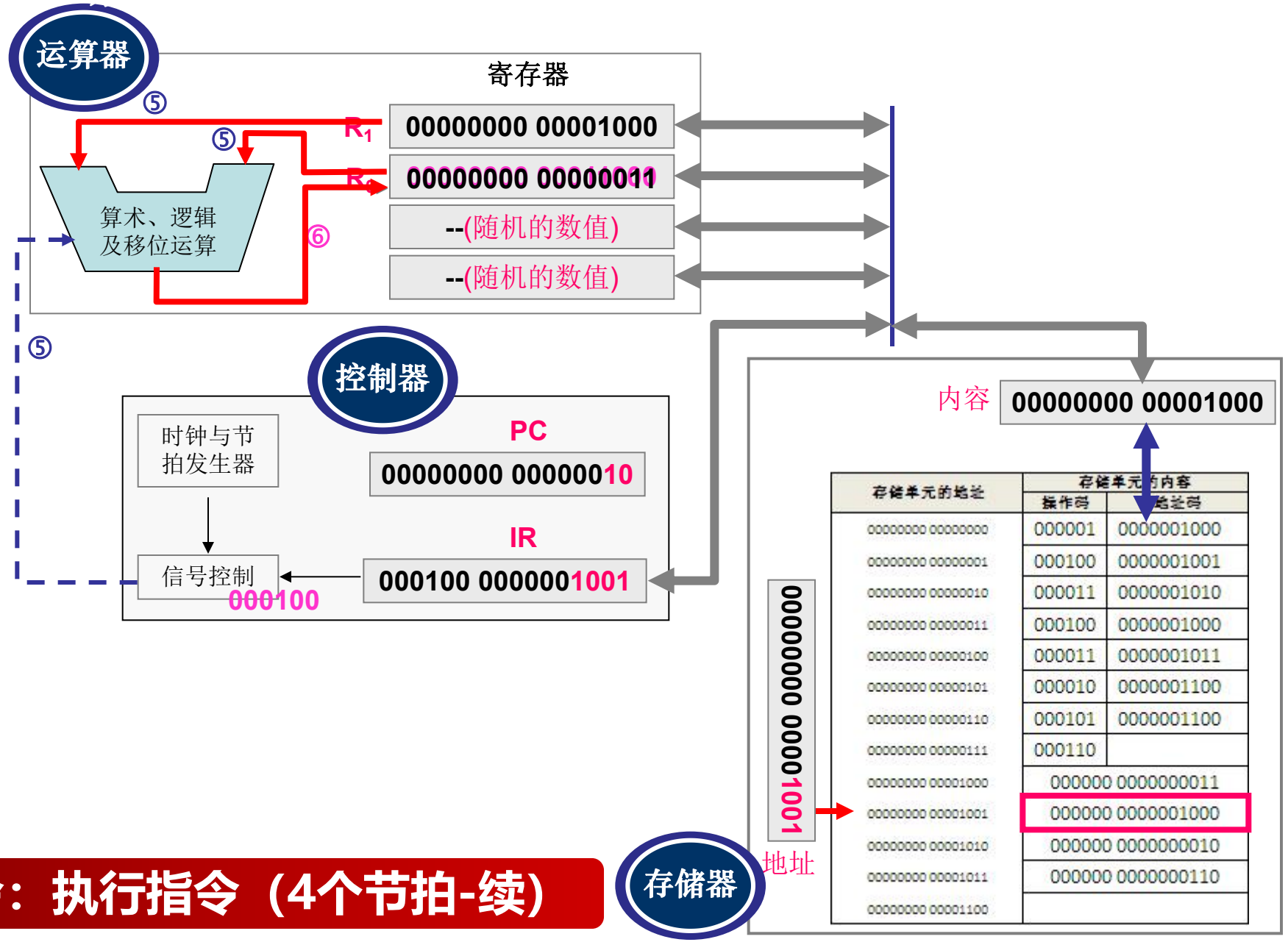
第2条指令：取指令（2个节拍）

机器程序的执行过程模拟



第2条指令：执行指令（4个节拍）

机器程序的执行过程模拟



第2条指令：执行指令（4个节拍-续）

单选题 1分

设置

52

关于机器指令的执行，则下列说法不正确的是_____。

A 控制器不断地从存储器中读取指令，并按照指令的内容进行执行；

B 机器指令的执行即是在时钟节拍控制下产生一系列信号的过程；

C 没有时钟与节拍发生器，机器的指令也能正确地执行；

D 没有PC，机器就不能正确地执行程序

提交

程序是如何被执行的—小结

程序是如何被执行的—思维小结

机器级算法

计算 $8\times3^2+2\times3+6$ = $((8\times3)+2)\times3+6$

计算方法1	计算方法2
Step1: 取出数3至运算器中	Step1: 取出数3至运算器中
Step2: 乘以数3在运算器中	Step2: 乘以数8在运算器中
Step3: 乘以数8在运算器中	Step3: 加上数2在运算器中
Step4: 存结果8×3^2在存储器中	Step4: 乘以数3在运算器中
Step5: 取出数2至运算器中	Step5: 加上数6至运算器中
Step6: 乘以数3在运算器中	
Step7: 加上(8×3^2)在运算器中	
Step8: 加上数6在运算器中	

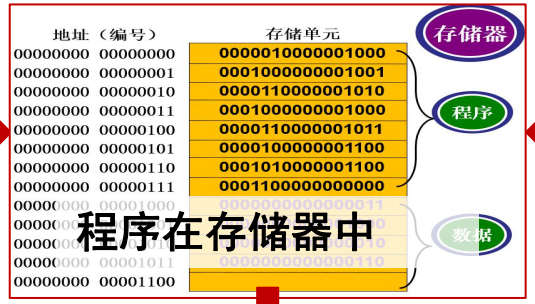
机器指令

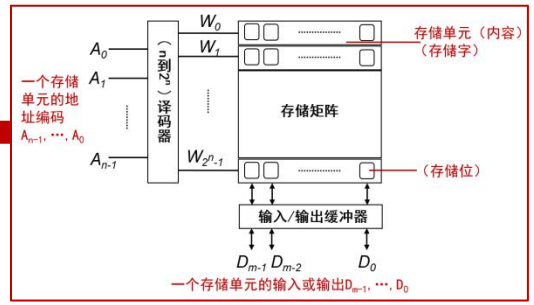
机器指令		对应的功能
操作码	地址码	
取数	α	α号存储单元的数 取出送到运算器；
000001	0000000100	
存数	β	运算器中的数 存储到 β号存储单元；
000010	0000010000	
加法	γ	运算器中的数 加上 γ号存储单元的数，结果保留在运算器；
000011	0000001010	
乘法	δ	运算器中的数 乘以 δ号存储单元的数，结果保留在运算器；
000100	0000001001	
打印		打印指令
000101	0000001100	
停机		停机指令
000110		

基本思维：算法程序化→程序指令化→指令存储化→执行信号化

机器程序

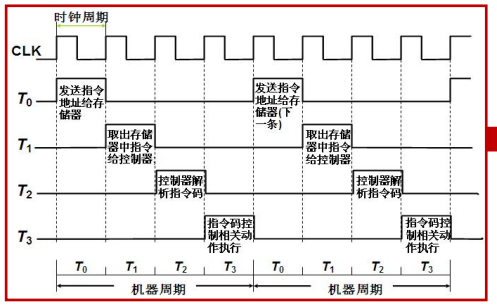
000001 0000001000
000100 0000001001
000011 0000001010
000100 0000001000
000011 0000001011
000010 0000001100
000101 0000001100
000110 0000000000

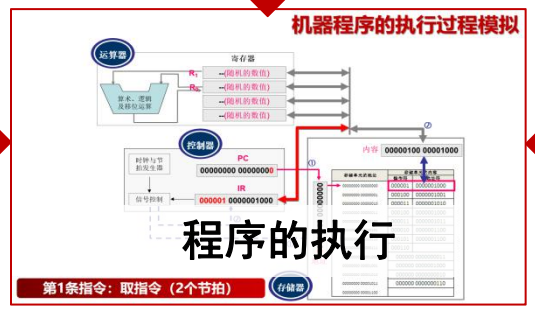


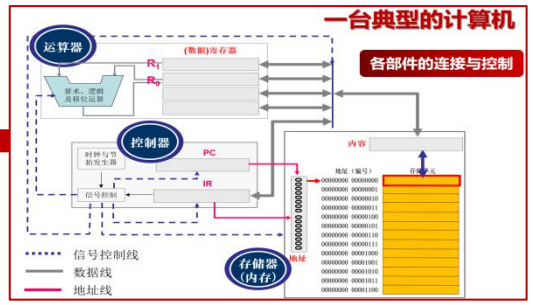


存储器

信号次序控制







典型计算机