中央处理器

功能

指令流

处理器执行的指令序列

数据流

根据指令操作要求依次存取数据的序列

控制器基本功能

对指令流（流出、分析与执行、流向）和数据流（流入与流出、数据变换、加工等操作）在时间与空间上实施正确的控制

主要寄存器

通用寄存器

存放原始数据和运算结果；可由CPU直接访问

专用寄存器

程序计数器（PC）

存放正在执行的**指令地址**或下条**指令地址**

顺序执行时，PC内容**不断增量（+1）**

指令寄存器（IR）

存放从**存储器**中**取出的指令**

存储器数据寄存器

暂存由主存储器读出的一条指令/一个数据字；向主存存入一条指令/一个数据字时，也暂存

存储器地址寄存器

保存当前CPU所访问的**主存单元地址**

状态标志寄存器

存放**程序状态字。**

程序状态字

程序状态字各位表征程序和机器运行的状态

状态标志&控制标志

例：

在CPU中**跟踪指令后继地址**的寄存器是___。

A. 主存地址寄存器 B. 程序计数器 **C. 指令寄存器** D. 程序状态字寄存器

指令寄存器的位数取决于_。

A. 存储器的容量 **B. 指令字长** C. 机器字长 D. 存储字长

在计算机系统中，表征系统运行状态的部件是_。

A. 程序计数器 B. 累加寄存器 C. 中断寄存器 **D. 程序状态字**

通用寄存器是_。

A. 可存放指令的寄存器 B. 可存放程序状态字的寄存器

C. 本身具有计数逻辑与移位逻辑的寄存器 **D. 可编程指定多种功能的寄存器**

组成

控制器

从主存中**取出一条指令**，并**指出下一条指令**在主存中的位置。

对指令进行**译码或测试**，产生相应的操作控制信号，以便启动规定的动作。

指挥并控制CPU、主存和输入/输出设备之间的**数据流动**方向。

运算器

执行所有的**算术运算**；

执行所有的**逻辑运算**，并进行逻辑测试。

技术参数

字长

单位时间内同时处理的二进制数据的位数（8位、32位、64位等）

内部工作频率

又称内频或主频，CPU内数字脉冲信号震荡的速度。

内频的倒数是**时钟周期**，CPU中最小的时间元素

外部工作频率

主板为CPU提供的基准时钟频率。内部倍频技术：**内频=外频×倍频**

前端总线频率

前端总线：CPU和外界交换数据的最主要通道

现阶段前端总线频率高于外频，QDR技术等达到2倍、4倍甚至更高

QPI数据传输速率

基于包传输的高速点到点连接技术，使用GT/s、MT/s为速率，T指transfer

例：QPI总线采取2:1倍率，时钟频率基于2.4GHz、3.2GHz，则数据传输速率为4.8GT/s、6.4GT/s。每次传输数据有16位有效数字，总带宽4.8GT/s×2Byte（16位）×2（双向）=19.2GB/s。

片内Cache容量

L1 Cache：位于CPU内核旁边，结合最紧密，分为一级数据缓存和一级指令缓存

L2 Cache：影响CPU性能关键因素之一

L3 Cache：进一步提高CPU效率

工作电压

地址总线宽度

决定CPU可访问的最大物理地址空间（即能够使用多大主存）

例：Pentium有32位地址线，可寻址最大2^32=4GB

数据总线宽度

决定CPU与外部Cache、主存及输入输出设备之间进行一次数据传输的信息量，指明芯片传输能力

制造工艺

线宽：芯片内电路与电路之间的距离

控制器组成

指令部件

完成取指令并分析指令，包括程序计数器（PC）、指令寄存器（IR）、指令译码器（ID）、地址形成部件

时序部件

产生一定的时序信号，包括脉冲源、启停控制逻辑、节拍信号发生器

微操作信号发生器

一条指令的取出和执行可以**分解为很多最基本的操作**，即微操作。微操作信号发生器也称为控制单元（CU）

中断控制逻辑

控制器硬件实现方法

组合逻辑性

称为常规控制器或硬布线控制器，微操作序列形成部件是由门电路组成的复杂树形网络。速度快，结构不完整。

存储逻辑型

称为微程序控制器，把微操作信号代码化，使每条机器指令转化成为一段微程序并存入一个专门的存储器。易于实现自动化设计，速度比组合逻辑控制器慢。

组合逻辑和存储逻辑结合型

PLA控制器

时序系统

指令周期&机器周期

指令周期指**取指令、分析指令到执行完该指令**所需的全部时间

机器周期又称CPU周期，通常把**一条指令划分为若干个机器周期**，每个机器周期完成一个基本操作

指令周期 = i×机器周期

机器运行在不同的机器周期，其对应的周期状态触发器被置“1”，有且仅有一个触发器被置“1”

例：单周期处理器中所有指令的指令周期为一个时钟周期。下列关于单周期处理器的叙述中错误的是_。

A. 可以采用单总线结构数据通路（单总线结构一个周期不能干完所有工作）

B. 处理器时钟频率较低（考虑较慢指令）

C. 在指令执行过程中控制信号不变

D. 每条指令的CPI（一条指令执行所需周期数）为1

节拍

把一个机器周期分为若干个相等时间段，每个时间段对应一个电位信号，称为节拍电位信号

统一节拍器

采用统一的、具有**相等时间间隔和相同数目的节拍**，使得**所有机器周期等长**，又称定长CPU周期

分散节拍器

按机器周期**实际需要安排节拍数**，需要多少节拍就发出多少节拍，又称不定长CPU周期

延长节拍法

选择适当节拍数作为**基本节拍**，如果某机器周期内无法完成全部微操作，则可以**延长节拍**

时钟周期插入

一些微型机中，时序信号中**不设置节拍**，直接使用**时钟周期信号**

工作脉冲

节拍器内设置一个或几个工作脉冲，作为同步脉冲的来源。

多级时序系统

小型机中常采用机器周期、节拍、工作脉冲三级时序系统。每个机器周期中包括若干节拍，每个节拍内有一个脉冲。

控制方式

同步控制方式

各操作都由同一的时序信号控制，在每个机器周期中产生**统一数目的节拍电位和工作脉冲。**

设计简单、容易实现、有较多空闲时间

采用同步控制的目的是_。

A. 提高执行速度　B. 简化控制时序 **C. 满足不同操作对时间安排的需求** D. 满足不同设备对时间安排的需求

异步控制方式

根据指令或部件的具体情况决定，采用不同时序。

没有时间浪费，控制比较复杂。

异步控制常用于_。

A. CPU访问外围设备时 B. 微程序控制器中 C. CPU的内部控制中　D. 主存的内部控制中

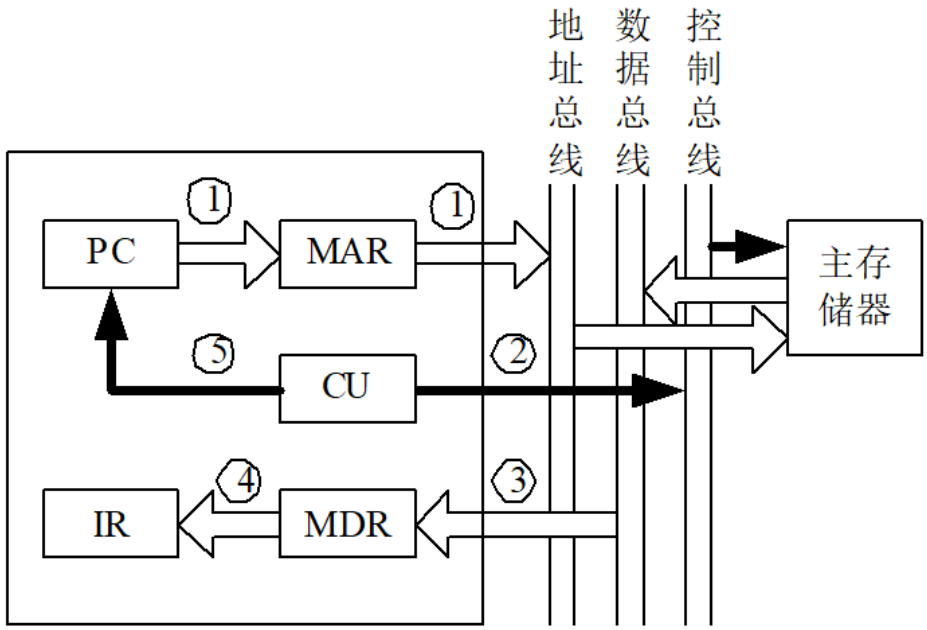
联合控制方法

指令执行基本过程

取指令阶段

将现行指令从主存储器中取出来并送至指令寄存器中去：

1. 将程序计数器（PC）中的内容送至存储器地址寄存器（MAR），并送地址总线（AB）。 **(PC)→MAR**
2. 由控制单元（CU）经控制总线（CB）向主存发读命令。 **Read**
3. 从主存中取出的指令通过数据总线（DB）送到存储器数据寄存器（MDR）。 **M(MAR) →MDR**
4. 将MDR的内容送至指令寄存器（IR）中。 **(MDR)→IR**
5. 将PC的内容递增，为取下一条指令做好准备。 **(PC) + 1→PC**



分析取数阶段

指令译码器ID可识别和区分不同的指令类型及各种获取操作数的方法

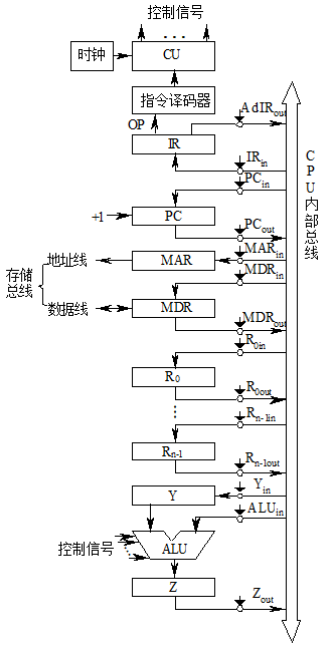
执行阶段

完成指令规定的各种操作，形成稳定的运算结果，并存储

指令的微操作序列

加法指令 ADD R1，@R0

把R0的内容作为地址送到主存以取得第一操作数，再与R1的内容相加，最后结果送回主存。即实现**((R0)) + (R1)→(R0)**



取指周期

1. PCout和MARin有效，完成**PC经CPU内部总线送至MAR**的操作，记作**(PC)→MAR**；
2. 通过控制总线（图中未画出）向主存发读命令，记作**Read**；
3. 存储器通过数据总线将MAR所指单元的内容（指令）送至MDR，记作**M(MAR)→MDR**；
4. MDRout和IRin有效，将MDR的内容送至IR，记作**(MDR)→IR。**至此，指令被从主存中取出，其操作码字段开始控制CU。
5. 使PC内容加1，记作**(PC)+1→PC。**

这条指令的微操作序列的第①～⑤步为取指令阶段的公共操作，它完成的任务为：

(PC)→MAR

Read

M(MAR)→MDR→IR

(PC) + 1→PC

取数周期

完成取操作数的任务，被加数在主存中，加数已放在通用寄存器R1中

1. R0out和MARin有效，完成将被加数地址送至MAR的操作，记作**(R0)→MAR**；
2. 向主存发读命令，记作**Read**；

3. 存储器通过数据总线将MAR所指单元的内容（数据）送至MDR，同时MDRout和Yin有效，记作**M(MAR)→MDR→Y**；

执行周期

完成加法运算任务，并将结果写回主存

1. R1out和ALUin有效，同时**CU向ALU发“ADD”控制信号**，使R1的内容和Y的内容相加，结果送寄存器Z中，记作**(R1)+Y→Z**；
2. Zout和MDRin有效，将运算结果送MDR，记作**(Z)→MDR。**
3. 主存发写命令，记作**Write。**

转移指令 JC A

若**上次运算结果有进位（C=1）**，就**转移**；若上次运算结果无进位（C=0），就顺序执行下一条指令。设A为位移量，转移地址等于PC的内容加位移量。

取指周期

与上条指令相同

执行周期

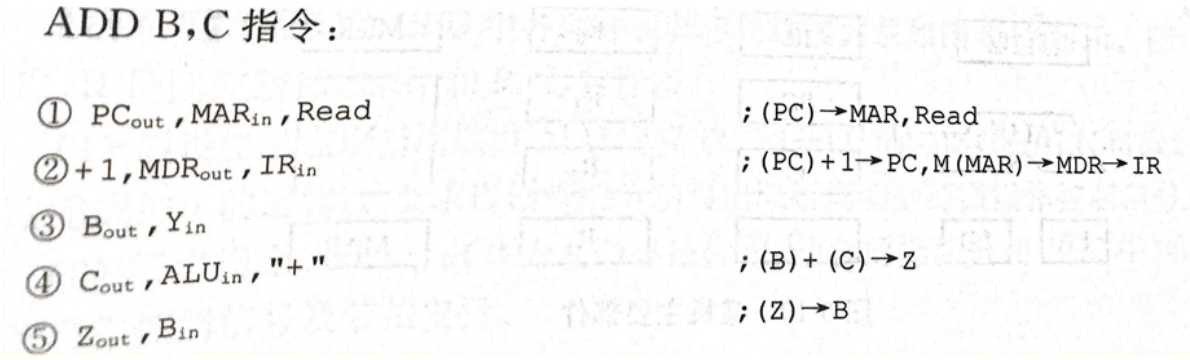
如果**有进位（C=1）**，则完成**(PC)+A→PC**的操作，否则跳过以下几步。

1. PCout和Yin有效，记作**(PC)→Y（C=1）**；
2. Ad IRout和ALUin有效，同时CU向ALU发“ADD”控制信号，使IR中的地址码字段A和Y的内容相加，结果送寄存器Z，记作**Ad(IR)+Y→Z （C=1）**；
3. Zout和PCin有效，将运算结果送PC，记作**(Z)→PC （C=1）**。

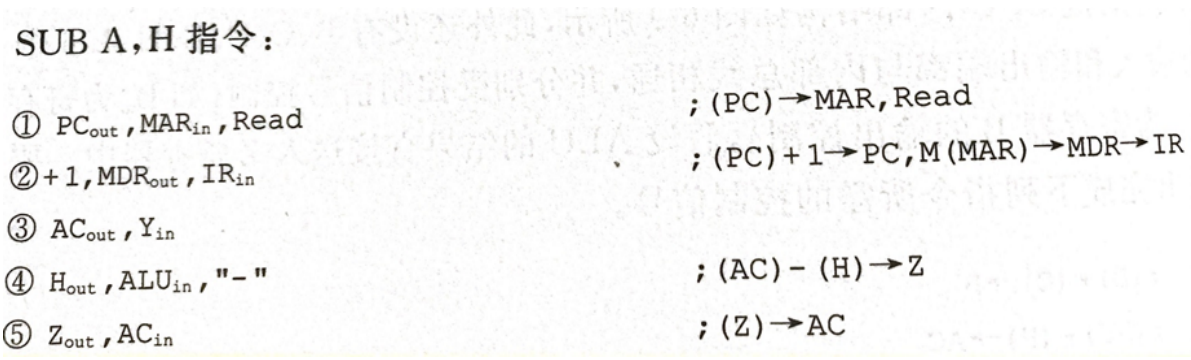
例

设某CPU内部结构如图所示，此外还设有B、C、D、E、H、L 6个寄存器，它们各自的输入输出端都与内部总线相通，并分别受控制器（如Bin为寄存器B的输入控制， Bout为寄存器B的输出控制），假设ALU的结果直接送入Z寄存器中。要求从取指令开始，写出完成下列指令所需的控制信号。

ADD B,C；(B)+(C)→B



SUB A,H；(AC)-(H)→AC



冯·诺依曼机中指令和数据均以二进制形式存放在存储器中，CPU区分它们的依据是(**指令周期的不同阶段**)。

指令译码器的作用是对进行译码。

A. 整条指令　　**B. 指令的操作码字段**　C. 指令的地址　　D. 指令的操作数字段

指令周期（**大于**/等于/小于）机器周期

微程序控制原理

基本概念

微命令和微操作

微操作是计算机中**最基本的、不可再分解的操作**；微命令是控制计算机各部件完成某个**基本微操作的命令。**

微命令与微操作一一对应，前者是后者的控制信号，后者是前者的操作过程。

微命令有兼容性和互斥性之分。

下列叙述中正确的是，同一CPU周期中，_。

A. 可以并行执行的操作称为兼容性微操作。

B. 不可以并行执行的操作称为兼容性微操作。

C. 可以并行执行的操作称为互斥性微操作。

D. 不可以并行执行的操作称为互斥性微操作。

微指令和微地址

微指令指控制存储器中一个单元的内容，即控制字，是**若干个微命令的集合。**包含操作控制字段（用以产生某步操作所需各微操作控制信号）和顺序控制字段（控制产生下一条要执行的微指令地址）

存放控制字的控制存储器的单元地址就称为微地址。

微周期

从控制存储器中读取一条微指令并执行相应的微命令所需全部时间。

微程序

一系列微指令的有序集合就是微程序。**一条机器指令对应于一段微程序。**

程序：指令的集合→指令（微程序）：微指令的集合→微指令：微命令的集合→微命令

微指令编码法

直接控制法（不译码法）

字段的每个独立二进制位代表一个微命令，1为有效，0为无效

结构简单，并行性强，操作速度快，微指令字太长

最短编码法

每条微指令只定义一个微命令。操作控制字段长度L≥log(2)N，N为微命令总数

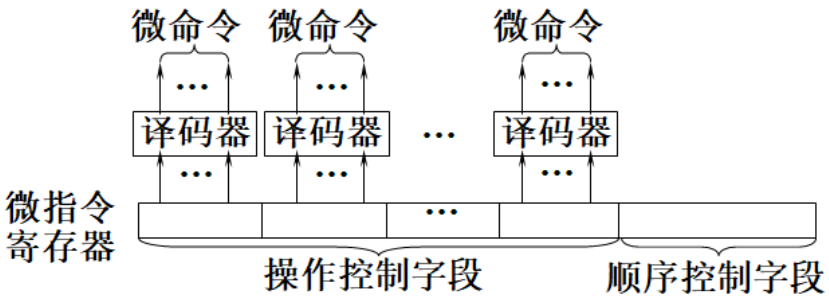
微指令字长最短，但需要通过一个微命令译码器译码后才能得到需要的微命令

字段编码法

将操作控制字段分为若干小段，每段内采用最短编码法，段间采用直接控制法。

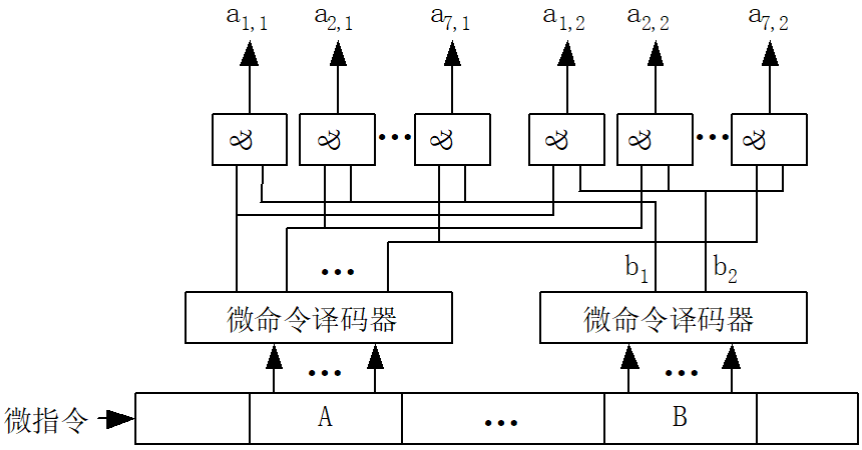
字段直接编码法

各字段都可以独立地定义本字段的微命令，而和其他字段无关。



字段间接编码法

一个字段的某些编码不能独立地定义某些微命令，而需要与其他字段的编码来联合定义



分段原则

1. 把**互斥性的微命令分在同一段内，兼容性的微命令分在不同段内。**
2. 应与数据通路结构相适应。
3. 每个小段中包含的**信息位不能太多**，否则将增加译码线路的复杂性和译码时间。
4. 一般**每个小段还要留出一个状态，表示本字段不发出任何微命令。**因此当某字段的长度为三位时，最多只能表示七个互斥的微命令，通常用000表示不操作。

例：某计算机的控制器采用微程序控制方式，微指令中的操作控制字段采用直接编码法，共有33个微命令，构成5个互斥类，分别包含7、3、12、5和6个微命令，则操作控制字段至少有位。

3+2+4+3+3=15

微程序控制器的基本组成

相比组合逻辑控制器多出**控制存储器（CM，存放微程序）**、**微指令寄存器（μIR，存放从CM取出的微指令）**、**微地址形成部件（产生初始微地址和后继微地址）**、**微地址寄存器（μMAR，接受微地址形成部件送来的微地址，为CM中读取微指令做准备）**

微程序控制器的工作过程

1. 执行**取指令公操作。**具体的执行是：在机器开始运行时，自动**将取指微程序的入口微地址送μMAR**，并**从CM中读出相应的微指令送入μIR。**微指令的操作控制字段**产生有关的微命令**，用来控制实现取机器指令的公共操作。取指微程序的入口地址一般为**CM的0号单元**，当取指微程序执行完后，从主存中取出的机器指令就已存入**指令寄存器IR**中了。
2. 由机器指令的**操作码字段**通过**微地址**形成部件产生出该机器指令所对应的**微程序的入口地址**，并**送入μMAR。**
3. 从CM中**逐条取出对应的微指令并执行之。**
4. 执行完对应于一条机器指令的一段微程序后又**回到取指微程序的入口地址**，继续第(1)步，以完成取下条机器指令的公共操作。

例：相对于微程序控制器，硬布线控制器的特点是，指令执行速度（**快**/慢），指令功能的修改和扩展（容易/**难**）。

微程序入口地址的形成

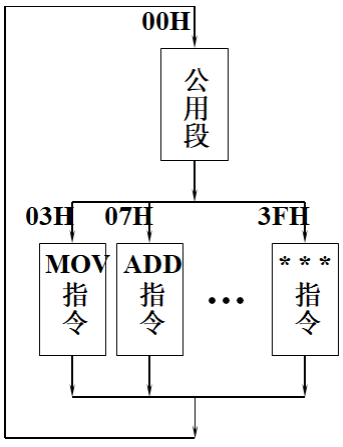
每条机器指令对应一段微程序，当公用的取指微程序从主存中取出机器指令之后，**由机器指令的操作码字段指出各段微程序的入口地址（初始微地址）。**

一级功能转换

如果机器指令**操作码字段的位数和位置固定**，可以**直接使操作码与入口地址码的部分位相对应。**

例如，某机有16条机器指令，指令操作码由4位二进制数表示，现以字母θ表示操作码，令微程序的**入口地址**为：θ11B

执行完后为0BH，各微程序的**入口地址相差4个单元。**



二级功能转换

不同类机器指令的**操作码的位数和位置不固定**，不能再采用一级功能转换的方法。

第一次**先按指令类型标志转移**，以区分出指令属于哪一类，如：是单操作数指令，还是双操作数指令等。因为每一类机器指令中操作码字段的位数和位置是固定的，所以**第二次即可按操作码区分出具体是哪条指令**，以便找出**相应微程序的入口微地址。**

通过PLA电路实现功能转换

采用**PLA电路**将每条机器指令的操作码翻译成对应的微程序入口地址

后继微地址形成

每条微指令执行完毕都要根据要求形成后继微地址

增量方式（顺序 - 转移型微地址）

顺序执行时**后继微地址就是现行微地址加上一个增量（通常为1）**；转移或转子时，由微指令的顺序控制字段产生转移微地址。因此，在微程序控制器中应当有一个**微程序计数器（μPC）**，为了降低成本，一般情况下都是**将微地址寄存器mMAR改为具有计数功能的寄存器**，以代替μPC。

断定方式

断定方式的**后继微地址**可由**微程序设计者指定**，或者根据微指令所规定的**测试结果直接决定**后继微地址的全部或部分值。

这是一种直接给定与测试断定相结合的方式，其顺序控制字段一般由两部分组成：非测试段和测试段。

(1)**非测试段**，可由设计者指定，一般是微地址的高位部分，用来指定后继微地址在CM中的某个区域内。

(2)**测试段**，根据有关状态的测试结果确定其地址值，一般对应微地址的低位部分。这相当于在指定区域内断定具体的分支。所依据的测试状态可能是指定的**开关状态、指令操作码、状态字**等。

测试段如果只有一位，则微地址将产生两个分支，若有两位，则最多可产生四个分支，依此类推，测试段为n位最多可产生2^n个分支。

下列说法中正确的是_.

A. 采用微程序控制器是为了提高速度

B. 控制存储器由高速RAM电路组成（前者在CPU）

C. 微指令计数器决定指令执行顺序（指令计数器决定）

D. 一条微指令放在控制存储器的一个单元中

下列说法中正确的是。

A. 微程序控制方式与硬布线控制方式相比较，前者可以使指令的执行速度更快。

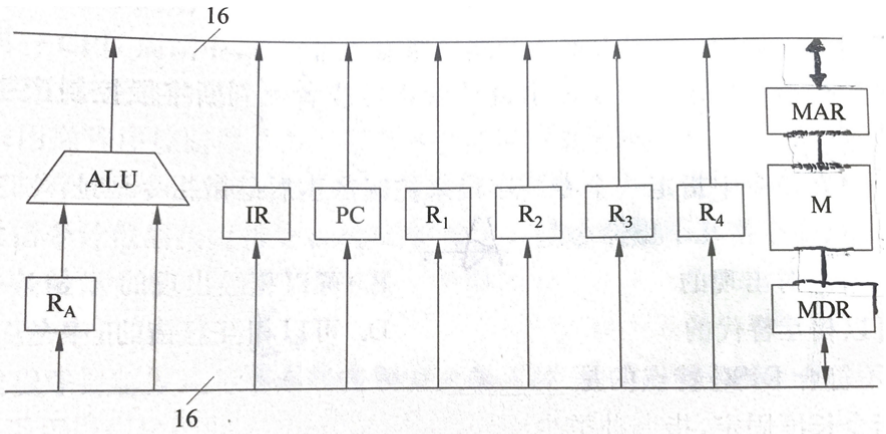
B. 若采用微程序控制方式，则可用μPC取代PC

C. 控制存储器可以用掩膜ROM、EPROM或闪速存储器实现

D. 指令周期也称为CPU周期

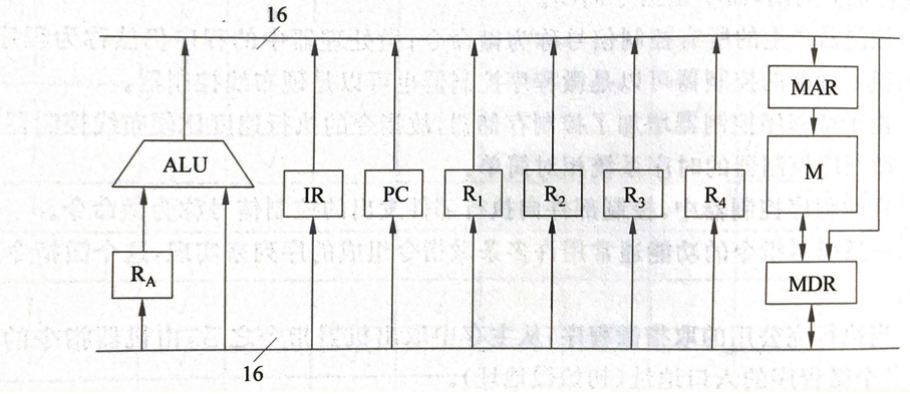
一个CPU的数据通路为双总线结构，如图所示。其中的连线有误。回答下列问题：

1. 画出修正错误后的数据通路结构，不能改变原有的双总线结构。
2. 如果要实现直接寻址方式，如何修改?
3. 描述ADD addr,R1指令从取指令开始的实现过程。指令功能为(R1)+(addr)→addr。



注：ALU—运算器；R_A—ALU 的输入寄存器；IR—指令寄存器；
PC—程序计数器；R_1～R_4—程序员可用的通用寄存器；
MAR—存储器地址寄存器；MDR—存储器数据寄存器

1.（1）修改后的数据通路结构如图 6-36 所示。



（2）直接寻址，IRaddr→MAR，有通路，不用修改

（3）指令 ADD addr,R_1 的实现过程如下：

PC→MAR	;从存储器中取指令
M(MAR)→MDR	
MDR→IR	
PC+1→PC	
IR_{addr}→MAR	;从存储器中取加数
M(MAR)→MDR	
MDR→R_A	
R_1→MDR	;从寄存器 R_1 中取被加数
+,ALU→MDR	;求和
MDR→M	;和写回存储器

PC不能被具有加“1”功能的存储器地址寄存器代替，因为MAR除存放指令地址还存放数据地址，而微程序控制器中μPC只存放微指令的地址。

某机有8条微指令I1~I8，每条微指令所含的微命令控制信号如表所示。为这10条微命令设计格式并安排编码。

微指令	微命令控制信号									
	a	b	c	d	e	f	g	h	i	j
I_1	√			√						
I_2			√				√		√	
I_3		√				√		√		
I_4	√									√
I_5			√		√				√	
I_6	√			√						√
I_7	√		√							
I_8		√				√		√		

解：bcd, efghij互斥，

2位	2位	2位	1位
00：不操作	00：不操作	00：不操作	0：不操作
01：b	01：e	01：h	1：a
10：c	10：f	10：i	
11：d	11：g	11：j	

I1: 11 00 00 1 I2: 10 11 10 0

I3: 01 10 01 0 I4: 00 00 11 1

I5: 10 01 10 0 I6: 11 00 11 1

I7: 10 00 00 1 I8: 01 10 01 0