第一章 计算机系统概论

1. 什么是计算机系统、计算机硬件和计算机软件？硬件和软件哪个更重要？

解：P3

计算机系统：由计算机硬件系统和软件系统组成的综合体。

计算机硬件：指计算机中的电子线路和物理装置。

计算机软件：计算机运行所需的程序及相关资料。

硬件和软件在计算机系统中相互依存，缺一不可，因此同样重要。

5. 冯•诺依曼计算机的特点是什么？

解：冯•诺依曼计算机的特点是：P8

- 计算机由运算器、控制器、存储器、输入设备、输出设备五大部件组成；
- 指令和数据以同同等地位存放于存储器内，并可以按地址访问；
- 指令和数据均用二进制表示；
- 指令由操作码、地址码两大部分组成，操作码用来表示操作的性质，地址码用来表示操作数在存储器中的位置；
- 指令在存储器中顺序存放，通常自动顺序取出执行；
- 机器以运算器为中心（原始冯•诺依曼机）。

7. 解释下列概念：

主机、CPU、主存、存储单元、存储元件、存储基元、存储元、存储字、存储字长、存储容量、机器字长、指令字长。

解：P9-10

主机：是计算机硬件的主体部分，由 CPU 和主存储器 MM 合成为主机。

CPU：中央处理器，是计算机硬件的核心部件，由运算器和控制器组成；（早期的运算器和控制器不在同一芯片上，现在的 CPU 内除含有运算器和控制器外还集成了 CACHE）。

主存：计算机中存放正在运行的程序和数据的存储器，为计算机的主要工作存储器，可随机存取；由存储体、各种逻辑部件及控制电路组成。

存储单元：可存放一个机器字并具有特定存储地址的存储单位。

存储元件：存储一位二进制信息的物理元件，是存储器中最小的存储单位，又叫存储基元或存储元，不能单独存取。

存储字：一个存储单元所存二进制代码的逻辑单位。

存储字长：一个存储单元所存二进制代码的位数。

存储容量：存储器中可存二进制代码的总量；（通常主、辅存容量分开描述）。

机器字长：指 CPU 一次能处理的二进制数据的位数，通常与 CPU 的寄存器位数有关。

指令字长：一条指令的二进制代码位数。

8. 解释下列英文缩写的中文含义：

CPU、PC、IR、CU、ALU、ACC、MQ、X、MAR、MDR、I/O、MIPS、CPI、FLOPS

解：全面的回答应分英文全称、中文名、功能三部分。

CPU：Central Processing Unit，中央处理机（器），是计算机硬件的核心部件，主要由运算器和控制器组成。

PC：Program Counter，程序计数器，其功能是存放当前欲执行指令的地址，并可自动

计数形成下一条指令地址。

IR：Instruction Register，指令寄存器，其功能是存放当前正在执行的指令。

CU：Control Unit，控制单元（部件），为控制器的核心部件，其功能是产生微操作命令序列。

ALU：Arithmetic Logic Unit，算术逻辑运算单元，为运算器的核心部件，其功能是进行算术、逻辑运算。

ACC：Accumulator，累加器，是运算器中既能存放运算前的操作数，又能存放运算结果的寄存器。

MQ：Multiplier-Quotient Register，乘商寄存器，乘法运算时存放乘数、除法时存放商的寄存器。

X：此字母没有专指的缩写含义，可以用作任一部件名，在此表示操作数寄存器，即运算器中工作寄存器之一，用来存放操作数；

MAR：Memory Address Register，存储器地址寄存器，在主存中用来存放欲访问的存储单元的地址。

MDR：Memory Data Register，存储器数据缓冲寄存器，在主存中用来存放从某单元读出、或要写入某存储单元的数据。

I/O：Input/Output equipment，输入/输出设备，为输入设备和输出设备的总称，用于计算机内部和外界信息的转换与传送。

MIPS：Million Instruction Per Second，每秒执行百万条指令数，为计算机运算速度指标的一种计量单位。

9. 画出主机框图，分别以存数指令“STA M”和加法指令“ADD M”（M 均为主存地址）为例，在图中按序标出完成该指令（包括取指令阶段）的信息流程（如→①）。假设主存容量为 256M*32 位，在指令字长、存储字长、机器字长相等的条件下，指出图中各寄存器的位数。

解：主机框图如 P13 图 1.11 所示。

（1）STA M 指令：PC→MAR，MAR→MM，MM→MDR，MDR→IR，

OP(IR) →CU，Ad(IR) →MAR，ACC→MDR，MAR→MM，WR

（2）ADD M 指令：PC→MAR，MAR→MM，MM→MDR，MDR→IR，

OP(IR) →CU，Ad(IR) →MAR，RD，MM→MDR，MDR→X，ADD，ALU→ACC，ACC→MDR，WR

假设主存容量 256M*32 位，在指令字长、存储字长、机器字长相等的条件下，ACC、X、IR、MDR 寄存器均为 32 位，PC 和 MAR 寄存器均为 28 位。

10. 指令和数据都存于存储器中，计算机如何区分它们？

解：计算机区分指令和数据有以下 2 种方法：

- 通过不同的时间段来区分指令和数据，即在取指令阶段（或取指微程序）取出的为指令，在执行指令阶段（或相应微程序）取出的即为数据。
- 通过地址来源区分，由 PC 提供存储单元地址的取出的是指令，由指令地址码部分提供存储单元地址的取出的是操作数。

第 2 章 计算机的发展及应用

1. 通常计算机的更新换代以什么为依据？

答：P22

主要以组成计算机基本电路的元器件为依据，如电子管、晶体管、集成电路等。

2. 举例说明专用计算机和通用计算机的区别。

答：按照计算机的效率、速度、价格和运行的经济性和实用性可以将计算机划分为通用计算机和专用计算机。通用计算机适应性强，但牺牲了效率、速度和经济性，而专用计算机是最有效、最经济和最快的计算机，但适应性很差。例如个人电脑和计算器。

3. 什么是摩尔定律？该定律是否永远生效？为什么？

答：P23，否，P36

第 3 章 系统总线

1. 什么是总线？总线传输有何特点？为了减轻总线负载，总线上的部件应具备什么特点？
答：P41.总线是多个部件共享的传输部件。

总线传输的特点是：某一时刻只能有一路信息在总线上传输，即分时使用。

为了减轻总线负载，总线上的部件应通过三态驱动缓冲电路与总线连通。

4. 为什么要设置总线判优控制？常见的集中式总线控制有几种？各有何特点？哪种方式响应时间最快？哪种方式对电路故障最敏感？
答：总线判优控制解决多个部件同时申请总线时的使用权分配问题；

常见的集中式总线控制有三种：链式查询、计数器定时查询、独立请求；

特点：链式查询方式连线简单，易于扩充，对电路故障最敏感；计数器定时查询方式优先级设置较灵活，对故障不敏感，连线及控制过程较复杂；独立请求方式速度最快，但硬件器件用量大，连线多，成本较高。

5. 解释下列概念：总线宽度、总线带宽、总线复用、总线的主设备（或主模块）、总线的从设备（或从模块）、总线的传输周期和总线的通信控制。
答：P46。

总线宽度：通常指数据总线的根数；

总线带宽：总线的数据传输率，指单位时间内总线上传输数据的位数；

总线复用：指同一条信号线可以分时传输不同的信号。

总线的主设备（主模块）：指一次总线传输期间，拥有总线控制权的设备（模块）；

总线的从设备（从模块）：指一次总线传输期间，配合主设备完成数据传输的设备（模块），它只能被动接受主设备发来的命令；

总线的传输周期：指总线完成一次完整而可靠的传输所需时间；

总线的通信控制：指总线传送过程中双方的时间配合方式。

6. 试比较同步通信和异步通信。
答：同步通信：指由统一时钟控制的通信，控制方式简单，灵活性差，当系统中各部件工作速度差异较大时，总线工作效率明显下降。适合于速度差别不大的场合。

异步通信：指没有统一时钟控制的通信，部件间采用应答方式进行联系，控制方式较同步复杂，灵活性高，当系统中各部件工作速度差异较大时，有利于提高总线工作效率。

8. 为什么说半同步通信同时保留了同步通信和异步通信的特点？
答：半同步通信既能像同步通信那样由统一时钟控制，又能像异步通信那样允许传输时间不一致，因此工作效率介于两者之间。

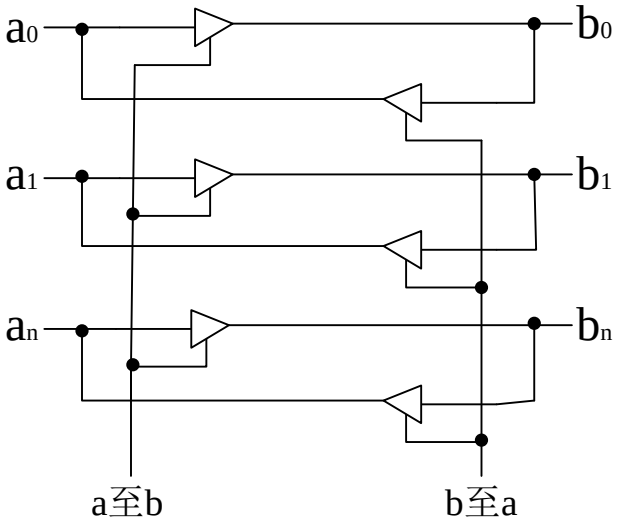
10. 为什么要设置总线标准？你知道目前流行的总线标准有哪些？什么叫 plug and play？哪些总线有这一特点？
答：总线标准的设置主要解决不同厂家各类模块化产品的兼容问题；

目前流行的总线标准有：ISA、EISA、PCI 等；

plug and play：即插即用，EISA、PCI 等具有此功能。

11. 画一个具有双向传输功能的总线逻辑图。

答：在总线的两端分别配置三态门，就可以使总线具有双向传输功能。



12. 设数据总线上接有 A、B、C、D 四个寄存器，要求选用合适的 74 系列芯片，完成下列逻辑设计：

（1） 设计一个电路，在同一时间实现 D→A、D→B 和 D→C 寄存器间的传送；

（2） 设计一个电路，实现下列操作：

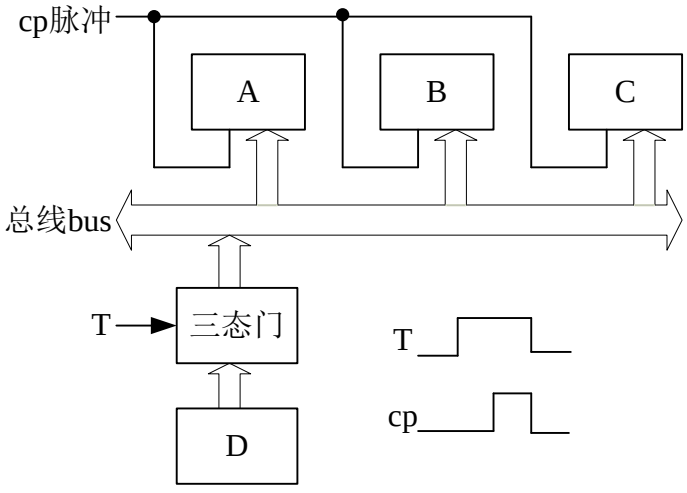
T0 时刻完成 D→总线；

T1 时刻完成总线→A；

T2 时刻完成 A→总线；

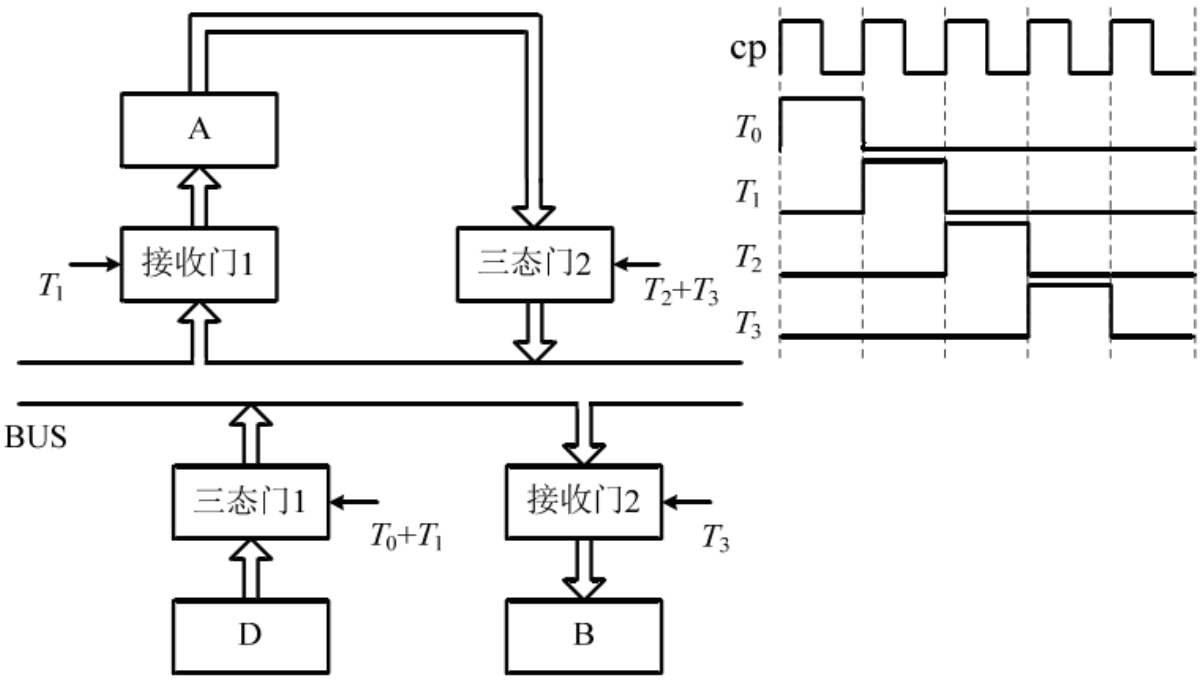
T3 时刻完成总线→B。

解：（1）由 T 打开三态门将 D 寄存器中的内容送至总线 bus，由 cp 脉冲同时将总线上的数据打入到 A、B、C 寄存器中。 T 和 cp 的时间关系如图（1）所示。



图（1）

（2）三态门 1 受 T0＋T1 控制，以确保 T0 时刻 D→总线，以及 T1 时刻总线→接收门 1→A。三态门 2 受 T2＋T3 控制，以确保 T2 时刻 A→总线，以及 T3 时刻总线→接收门 2→B。T0、T1、T2、T3 波形图如图（2）所示。



图(2)

第 四 章

3. 存储器的层次结构主要体现在什么地方？为什么要分这些层次？计算机如何管理这些层次？

答：存储器的层次结构主要体现在 Cache-主存和主存-辅存这两个存储层次上。

Cache-主存层次在存储系统中主要对 CPU 访存起加速作用，即从整体运行的效果分析，CPU 访存速度加快，接近于 Cache 的速度，而寻址空间和位价却接近于主存。

主存-辅存层次在存储系统中主要起扩容作用，即从程序员的角度看，他所使用的存储器其容量和位价接近于辅存，而速度接近于主存。

综合上述两个存储层次的作用，从整个存储系统来看，就达到了速度快、容量大、位价低的优化效果。

主存与 CACHE 之间的信息调度功能全部由硬件自动完成。而主存与辅存层次的调度目前广泛采用虚拟存储技术实现，即将主存与辅存的一部分通过软硬结合的技术组成虚拟存储器，程序员可使用这个比主存实际空间（物理地址空间）大得多的虚拟地址空间（逻辑地址空间）编程，当程序运行时，再由软、硬件自动配合完成虚拟地址空间与主存实际物理空间的转换。因此，这两个层次上的调度或转换操作对于程序员来说都是透明的。

4. 说明存取周期和存取时间的区别。

解：存取周期和存取时间的主要区别是：存取时间仅为完成一次操作的时间，而存取周期不仅包含操作时间，还包含操作后线路的恢复时间。即：

存取周期 = 存取时间 + 恢复时间

5. 什么是存储器的带宽？若存储器的数据总线宽度为 32 位，存取周期为 200ns，则存储器的带宽是多少？

解：存储器的带宽指单位时间内从存储器进出信息的最大数量。

存储器带宽 = 1/200ns ×32 位 = 160M 位/秒 = 20MB/秒 = 5M 字/秒

注意：字长 32 位，不是 16 位。（注：$1ns=10^{-9}s$）

6. 某机字长为 32 位，其存储容量是 64KB，按字编址它的寻址范围是多少？若主存以字节编址，试画出主存字地址和字节地址的分配情况。

解：存储容量是 64KB 时，按字节编址的寻址范围就是 64K，如按字编址，其寻址范围为：

64K / （32/8）= 16K

主存字地址和字节地址的分配情况：（略）。

7. 一个容量为 16K×32 位的存储器，其地址线和数据线的总和是多少？当选用下列不同规格的存储芯片时，各需要多少片？

1K×4 位，2K×8 位，4K×4 位，16K×1 位，4K×8 位，8K×8 位

解：地址线和数据线的总和 = 14 + 32 = 46 根；

选择不同的芯片时，各需要的片数为：

1K×4：（16K×32） / （1K×4） = 16×8 = 128 片

2K×8：（16K×32） / （2K×8） = 8×4 = 32 片

4K×4：（16K×32） / （4K×4） = 4×8 = 32 片

16K×1：（16K×32）/ （16K×1） = 1×32 = 32 片

4K×8：（16K×32）/ （4K×8） = 4×4 = 16 片

8K×8：（16K×32） / （8K×8） = 2×4 = 8 片

8. 试比较静态 RAM 和动态 RAM。

答：略。（参看课件）

9. 什么叫刷新？为什么要刷新？说明刷新有几种方法。

解：刷新：对 DRAM 定期进行的全部重写过程；

刷新原因：因电容泄漏而引起的 DRAM 所存信息的衰减需要及时补充，因此安排了定期刷新操作；

常用的刷新方法有三种：集中式、分散式、异步式。

集中式：在最大刷新间隔时间内，集中安排一段时间进行刷新，存在 CPU 访存死时间。

分散式：在每个读/写周期之后插入一个刷新周期，无 CPU 访存死时间。

异步式：是集中式和分散式的折衷。

10. 半导体存储器芯片的译码驱动方式有几种？

解：半导体存储器芯片的译码驱动方式有两种：线选法和重合法。

线选法：地址译码信号只选中同一个字的所有位，结构简单，费器材；

重合法：地址分行、列两部分译码，行、列译码线的交叉点即为所选单元。这种方法通过行、列译码信号的重合来选址，也称矩阵译码。可大大节省器材用量，是最常用的译码驱动方式。

11. 一个 8K×8 位的动态 RAM 芯片，其内部结构排列成 256×256 形式，存取周期为 0.1μs。试问采用集中刷新、分散刷新和异步刷新三种方式的刷新间隔各为多少？

解：采用分散刷新方式刷新间隔为:2ms，其中刷新死时间为：256×0.1μs=25.6μs

采用分散刷新方式刷新间隔为：256×（0.1μs+×0.1μs）=51.2μs

采用异步刷新方式刷新间隔为:2ms

12. 画出用 1024×4 位的存储芯片组成一个容量为 64K×8 位的存储器逻辑框图。要求将 64K 分成 4 个页面，每个页面分 16 组，指出共需多少片存储芯片。

解：设采用 SRAM 芯片，则：

总片数 = （64K×8 位） / （1024×4 位）= 64×2 = 128 片

题意分析：本题设计的存储器结构上分为总体、页面、组三级，因此画图时也应分三级画。首先应确定各级的容量：

页面容量 = 总容量 / 页面数 =64K×8/4 =16K×8 位,4 片 16K×8 字串联成 64K×8 位

组容量 = 页面容量 / 组数 = 16K×8 位 / 16 = 1K×8 位，16 片 1K×8 位字串联成 16K×8 位

组内片数 = 组容量 / 片容量 = 1K×8 位 / 1K×4 位 = 2 片，两片 1K×4 位芯片位并联成 1K×8 位

存储器逻辑框图：（略）。

13. 设有一个64K×8位的RAM芯片，试问该芯片共有多少个基本单元电路(简称存储基元)？欲设计一种具有上述同样多存储基元的芯片，要求对芯片字长的选择应满足地址线和数据线

的总和为最小，试确定这种芯片的地址线和数据线，并说明有几种解答。

解：存储基元总数 = 64K×8 位 = 512K 位 = 2^{19}位；

思路：如要满足地址线和数据线总和最小，应尽量把存储元安排在字向，因为地址位数和字数成 2 的幂的关系，可较好地压缩线数。

解：设地址线根数为 a，数据线根数为 b，则片容量为：$2^a \times b = 2^{19}$；$b = 2^{19-a}$；

若 a = 19，b = 1，总和 = 19+1 = 20；

a = 18，b = 2，总和 = 18+2 = 20；

a = 17，b = 4，总和 = 17+4 = 21；

a = 16，b = 8，总和 = 16+8 = 24；

…… ……

由上可看出：片字数越少，片字长越长，引脚数越多。片字数减 1、片位数均按 2 的幂变化。

结论：如果满足地址线和数据线的总和为最小，这种芯片的引脚分配方案有两种：地址线 = 19 根，数据线 = 1 根；或地址线 = 18 根，数据线 = 2 根。

14. 某 8 位微型机地址码为 18 位，若使用 4K×4 位的 RAM 芯片组成模块板结构的存储器，试问：

（1）该机所允许的最大主存空间是多少？

（2）若每个模块板为 32K×8 位，共需几个模块板？

（3）每个模块板内共有几片 RAM 芯片？

（4）共有多少片 RAM?

（5）CPU 如何选择各模块板？

解：（1）该机所允许的最大主存空间是：2^{18} × 8 位 = 256K×8 位 = 256KB

（2）模块板总数 = 256K×8 / 32K×8 = 8 块

（3）板内片数 = 32K×8 位 / 4K×4 位 = 8×2 = 16 片

（4）总片数 = 16 片×8 = 128 片

（5）CPU 通过最高 3 位地址译码输出选择模板，次高 3 位地址译码输出选择芯片。地址格式分配如下：

模板号（3位）	芯片号 （3位）	片内地址（12位）

15. 设 CPU 共有 16 根地址线，8 根数据线，并用 $\overline{\text{MREQ}}$（低电平有效）作访存控制信号，$R/\overline{W}$ 作读写命令信号（高电平为读，低电平为写）。现有下列存储芯片：ROM（2K×8 位，4K×4 位，8K×8 位），RAM（1K×4 位，2K×8 位，4K×8 位），及 74138 译码器和其他门电路（门电路自定）。试从上述规格中选用合适芯片，画出 CPU 和存储芯片的连接图。要求：

（1）最小 4K 地址为系统程序区，4096~16383 地址范围为用户程序区；

（2）指出选用的存储芯片类型及数量；

（3）详细画出片选逻辑。

解：（1）地址空间分配图：

系统程序区（ROM 共 4KB）：0000H-0FFFH

用户程序区（RAM 共 12KB）：1000H-FFFFH

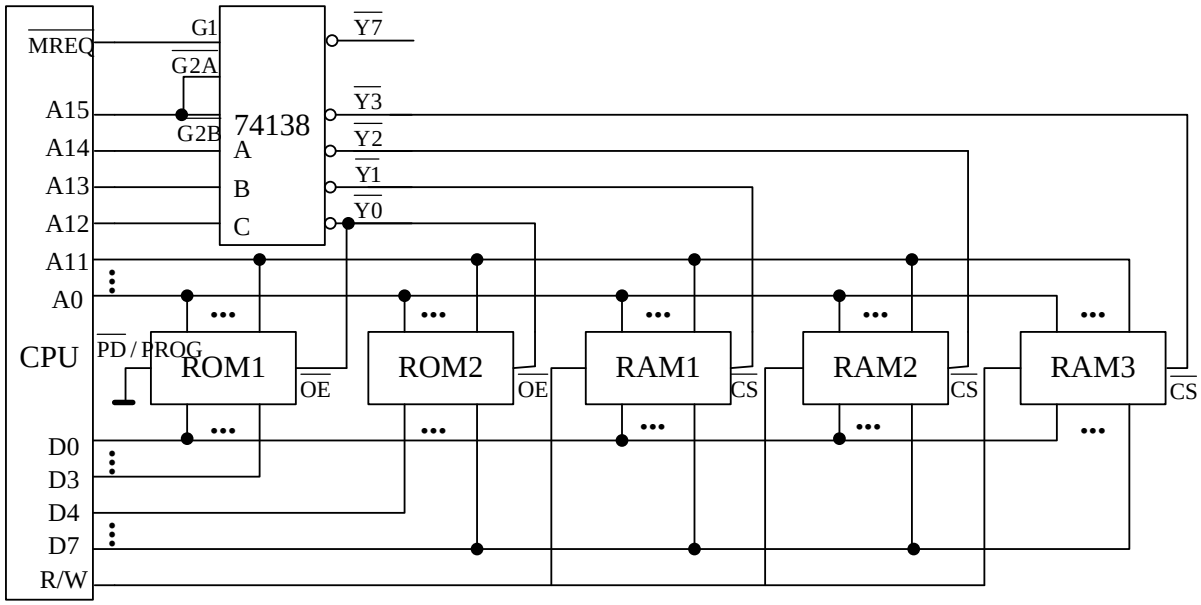
（2）选片：ROM：选择 4K×4 位芯片 2 片，位并联

RAM：选择 4K×8 位芯片 3 片，字串联(RAM1 地址范围为:1000H-1FFFH,RAM2 地址范围为 2000H-2FFFH, RAM3 地址范围为:3000H-3FFFH)

（3）各芯片二进制地址分配如下：

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
ROM1,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
RAM1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
RAM2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
RAM3	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CPU 和存储器连接逻辑图及片选逻辑如下图(3)所示：



图（3）

16. CPU 假设同上题，现有 8 片 8K×8 位的 RAM 芯片与 CPU 相连，试回答：

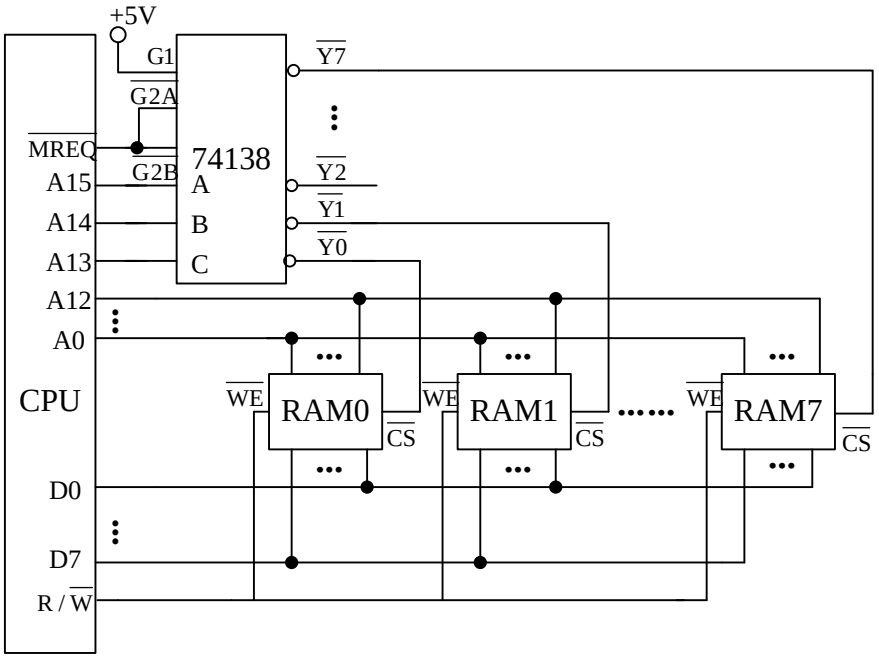
（1）用 74138 译码器画出 CPU 与存储芯片的连接图；

（2）写出每片 RAM 的地址范围；

（3）如果运行时发现不论往哪片 RAM 写入数据后，以 A000H 为起始地址的存储芯片都有与其相同的数据，分析故障原因。

（4）根据（1）的连接图，若出现地址线 A13 与 CPU 断线，并搭接到高电平上，将出现什么后果？

解：（1）CPU 与存储器芯片连接逻辑图：



（2）地址空间分配图：

RAM0:0000H-1FFFH

RAM1:2000H-3FFFH

RAM2:4000H-5FFFH

RAM3:6000H-7FFFH

RAM4:8000H-9FFFH

RAM5:A000H-BFFFH

RAM6:C000H-DFFFH

RAM7:E000H-FFFFH

（3)如果运行时发现不论往哪片RAM写入数据后,以A000H为起始地址的存储芯片(RAM5)都有与其相同的数据，则根本的故障原因为：该存储芯片的片选输入端很可能总是处于低电平。假设芯片与译码器本身都是好的，可能的情况有：

1）该片的-CS 端与-WE 端错连或短路；

2）该片的-CS 端与 CPU 的-MREQ 端错连或短路；

3）该片的-CS 端与地线错连或短路。

（4)如果地址线 A13 与 CPU 断线，并搭接到高电平上，将会出现 A13 恒为“1”的情况。此时存储器只能寻址 A13=1 的地址空间(奇数片)，A13=0 的另一半地址空间（偶数片）将永远访问不到。若对 A13=0 的地址空间（偶数片）进行访问，只能错误地访问到 A13=1 的对应空间(奇数片)中去。

17. 写出 1100、1101、1110、1111 对应的汉明码。

解：有效信息均为 n=4 位，假设有效信息用 b4b3b2b1 表示

校验位位数 k=3 位，（2^k>=n+k+1）

设校验位分别为 c1、c2、c3，则汉明码共 4+3=7 位，即：c1c2b4c3b3b2b1

校验位在汉明码中分别处于第 1、2、4 位

c1=b4⊕b3⊕b1

c2=b4⊕b2⊕b1

c3=b3⊕b2⊕b1

当有效信息为 1100 时，c3c2c1=011,汉明码为 1110100。

当有效信息为 1101 时，c3c2c1=100,汉明码为 0011101。

当有效信息为 1110 时，c3c2c1=101,汉明码为 1011110。

当有效信息为 1111 时，c3c2c1=010,汉明码为 0110111。

18. 已知收到的汉明码（按配偶原则配置）为 1100100、1100111、1100000、1100001，检查上述代码是否出错？第几位出错？

解：假设接收到的汉明码为：c1’ c2’ b4’ c3’ b3’ b2’ b1’

纠错过程如下：

P1=c1’ ⊕b4’ ⊕b3’ ⊕b1’

P2=c2’ ⊕b4’ ⊕b2’ ⊕b1’

P3=c3’ ⊕b3’ ⊕b2’ ⊕b1’

如果收到的汉明码为 1100100，则 p3p2p1=011，说明代码有错，第 3 位（b4’）出错，有效信息为：1100

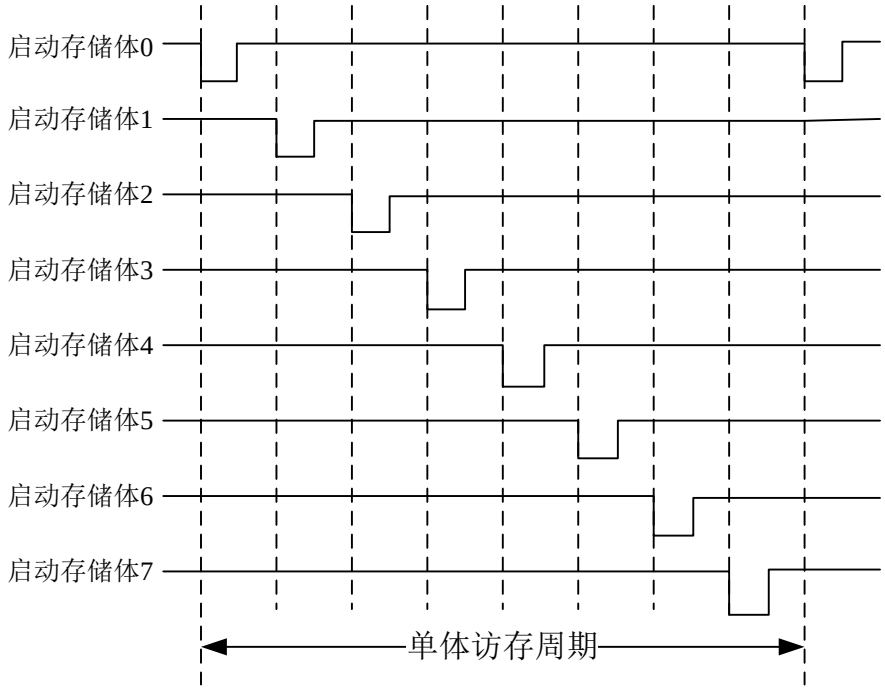
如果收到的汉明码为 1100111，则 p3p2p1=111，说明代码有错，第 7 位（b1’）出错，有效信息为：0110

如果收到的汉明码为 1100000，则 p3p2p1=110，说明代码有错，第 6 位（b2’）出错，有效信息为：0010

如果收到的汉明码为 1100001，则 p3p2p1=001，说明代码有错，第 1 位（c1’）出错，有效信息为：0001

22. 某机字长 16 位，常规的存储空间为 64K 字，若想不改用其他高速的存储芯片，而使访存速度提高到 8 倍，可采取什么措施？画图说明。

解：若想不改用高速存储芯片，而使访存速度提高到 8 倍，可采取八体交叉存取技术，8 体交叉访问时序如下图：



18. 什么是“程序访问的局部性”？存储系统中哪一级采用了程序访问的局部性原理？

解：程序运行的局部性原理指：在一小段时间内，最近被访问过的程序和数据很可能再次被访问；在空间上，这些被访问的程序和数据往往集中在一小片存储区；在访问顺序上，指令顺序执行比转移执行的可能性大（大约 5:1）。存储系统中 Cache—主存层次采用了程序访问的局部性原理。

25. Cache 做在 CPU 芯片内有什么好处？将指令 Cache 和数据 Cache 分开又有什么好处？

答：Cache 做在 CPU 芯片内主要有下面几个好处：

1）可提高外部总线的利用率。因为 Cache 在 CPU 芯片内，CPU 访问 Cache 时不必占用外部总线。

2）Cache 不占用外部总线就意味着外部总线可更多地支持 I/O 设备与主存的信息传输，增强了系统的整体效率。

3)可提高存取速度。因为 Cache 与 CPU 之间的数据通路大大缩短,故存取速度得以提高。

将指令 Cache 和数据 Cache 分开有如下好处：

1）可支持超前控制和流水线控制，有利于这类控制方式下指令预取操作的完成。

2）指令 Cache 可用 ROM 实现，以提高指令存取的可靠性。

3）数据 Cache 对不同数据类型的支持更为灵活，既可支持整数（例 32 位），也可支持浮点数据（如 64 位）。

补充：

Cache 结构改进的第三个措施是分级实现，如二级缓存结构，即在片内 Cache（L1）和主存之间再设一个片外 Cache（L2），片外缓存既可以弥补片内缓存容量不够大的缺点，又可在主存与片内缓存间起到平滑速度差的作用，加速片内缓存的调入调出速度。

30. 一个组相连映射的 CACHE 由 64 块组成，每组内包含 4 块。主存包含 4096 块，每块由 128 字组成，访存地址为字地址。试问主存和高速存储器的地址各为几位？画出主存地址格式。

解：cache 组数：64/4=16 ，Cache 容量为：$64*128=2^{13}$字，cache 地址 13 位

主存共分 4096/16=256 区，每区 16 块

主存容量为：$4096*128=2^{19}$字，主存地址 19 位，地址格式如下：

主存字块标记（8 位）	组地址（4 位）	字块内地址（7 位）

第 六 章

12. 设浮点数格式为：阶码 5 位（含 1 位阶符），尾数 11 位（含 1 位数符）。写出 51/128、-27/1024 所对应的机器数。要求如下：

（1）阶码和尾数均为原码。

（2）阶码和尾数均为补码。

（3）阶码为移码，尾数为补码。

解：据题意画出该浮点数的格式：

阶符 1 位	阶码 4 位	数符 1 位	尾数 10 位

将十进制数转换为二进制：x1= 51/128= 0.0110011B= 2^{-1} * 0.110 011B

x2= -27/1024= -0.0000011011B = 2^{-5}*(-0.11011B）

则以上各数的浮点规格化数为：

（1）[x1]浮=1，0001；0.110 011 000 0

[x2]浮=1，0101；1.110 110 000 0

（2）[x1]浮=1，1111；0.110 011 000 0

[x2]浮=1，1011；1.001 010 000 0

（3）[x1]浮=0，1111；0.110 011 000 0

[x2]浮=0，1011；1.001 010 000 0

16. 设机器数字长为 16 位，写出下列各种情况下它能表示的数的范围。设机器数采用一位符号位，答案均用十进制表示。

（1）无符号数；

（2）原码表示的定点小数。

（3）补码表示的定点小数。

（4）补码表示的定点整数。

（5）原码表示的定点整数。

（6）浮点数的格式为：阶码 6 位（含 1 位阶符），尾数 10 位（含 1 位数符）。分别写出其正数和负数的表示范围。

（7）浮点数格式同（6），机器数采用补码规格化形式，分别写出其对应的正数和负数的真值范围。

解：（1）无符号整数：0 —— 2^{16} - 1，即：0—— 65535；

无符号小数：0 —— 1 - 2^{-16} ，即：0 —— 0.99998；

（2）原码定点小数：-1 + 2^{-15}——1 - 2^{-15} ，即：-0.99997 —— 0.99997

（3）补码定点小数：- 1——1 - 2^{-15} ，即：-1——0.99997

（4）补码定点整数：-2^{15}——2^{15} - 1 ，即：-32768——32767

（5）原码定点整数：-2^{15} + 1——2^{15} - 1，即：-32767——32767

（6）据题意画出该浮点数格式，当阶码和尾数均采用原码，非规格化数表示时：

最大负数= 1，11 111；1.000 000 001 ，即 $-2^{-9}\times2^{-31}$

最小负数= 0，11 111；1.111 111 111，即 $-(1-2^{-9})\times2^{31}$

则负数表示范围为：$-(1-2^{-9})\times2^{31}$ —— $-2^{-9}\times2^{-31}$

最大正数= 0，11 111；0.111 111 111，即 $(1-2^{-9})\times2^{31}$

最小正数= 1，11 111；0.000 000 001，即 $2^{-9}\times2^{-31}$

则正数表示范围为：$2^{-9}\times2^{-31}$ ——$(1-2^{-9})\times2^{31}$

（7）当机器数采用补码规格化形式时，若不考虑隐藏位，则

最大负数=1，00 000；1.011 111 111，即 $-2^{-1}\times2^{-32}$

最小负数=0，11 111；1.000 000 000，即 -1×2^{31}

则负数表示范围为：-1×2^{31} —— $-2^{-1}\times2^{-32}$

最大正数=0，11 111；0.111 111 111，即 $(1-2^{-9})\times2^{31}$

最小正数=1，00 000；0.100 000 000，即 $2^{-1}\times2^{-32}$

则正数表示范围为：$2^{-1}\times2^{-32}$ ——$(1-2^{-9})\times2^{31}$

17. 设机器数字长为 8 位（包括一位符号位），对下列各机器数进行算术左移一位、两位，算术右移一位、两位，讨论结果是否正确。

[x1]原=0.001 1010；[y1]补=0.101 0100；[z1]反=1.010 1111；

[x2]原=1.110 1000；[y2]补=1.110 1000；[z2]反=1.110 1000；

[x3]原=1.001 1001；[y3]补=1.001 1001；[z3]反=1.001 1001。

解：算术左移一位：

[x1]原=0.011 0100；正确

[x2]原=1.101 0000；溢出（丢 1）出错

[x3]原=1.011 0010；正确

[y1]补=0.010 1000；溢出（丢 1）出错

[y2]补=1.101 0000；正确

[y3]补=1.011 0010；溢出（丢 0）出错

[z1]反=1.101 1111；溢出（丢 0）出错

[z2]反=1.101 0001；正确

[z3]反=1.011 0011；溢出（丢 0）出错

算术左移两位：

[x1]原=0.110 1000；正确

[x2]原=1.010 0000；溢出（丢 11）出错

[x3]原=1.110 0100；正确

[y1]补=0.101 0000；溢出（丢 10）出错

[y2]补=1.010 0000；正确

[y3]补=1.110 0100；溢出（丢 00）出错

[z1]反=1.011 1111；溢出（丢 01）出错

[z2]反=1.010 0011；正确

[z3]反=1.110 0111；溢出（丢 00）出错

算术右移一位：

[x1]原=0.000 1101；正确

[x2]原=1.011 0100；正确

[x3]原=1.000 1100(1)；丢 1，产生误差

[y1]补=0.010 1010；正确

[y2]补=1.111 0100；正确

[y3]补=1.100 1100(1)；丢 1，产生误差

[z1]反=1.101 0111；正确

[z2]反=1.111 0100(0)；丢 0，产生误差

[z3]反=1.100 1100；正确

算术右移两位：

[x1]原=0.000 0110（10）；产生误差
[x2]原=1.001 1010；正确
[x3]原=1.000 0110（01）；产生误差
[y1]补=0.001 0101；正确
[y2]补=1.111 1010；正确
[y3]补=1.110 0110（01）；产生误差
[z1]反=1.110 1011；正确
[z2]反=1.111 1010（00）；产生误差
[z3]反=1.110 0110（01）；产生误差

19. 设机器数字长为8位（含1位符号位），用补码运算规则计算下列各题。
（1）A=9/64， B=-13/32，求A+B。
（2）A=19/32，B=-17/128，求A-B。
（3）A=-3/16，B=9/32，求A+B。
（4）A=-87，B=53，求A-B。
（5）A=115，B=-24，求A+B。

解：（1）A=9/64= 0.001 0010B, B= -13/32= -0.011 0100B
[A]补=0.001 0010, [B]补=1.100 1100
[A+B]补= 0.0010010 + 1.1001100 = 1.1011110 ——无溢出
A+B= -0.010 0010B = -17/64

（2）A=19/32= 0.100 1100B, B= -17/128= -0.001 0001B
[A]补=0.100 1100, [B]补=1.110 1111 , [-B]补=0.001 0001
[A-B]补= 0.1001100 + 0.0010001= 0.1011101 ——无溢出
A-B= 0.101 1101B = 93/128B

（3）A= -3/16= -0.001 1000B, B=9/32= 0.010 0100B
[A]补=1.110 1000, [B]补= 0.010 0100
[A+B]补= 1.1101000 + 0.0100100 = 0.0001100 —— 无溢出
A+B= 0.000 1100B = 3/32

（4） A= -87= -101 0111B, B=53=110 101B
[A]补=1 010 1001, [B]补=0 011 0101, [-B]补=1 100 1011
[A-B]补= 1 0101001 + 1 1001011 = 0 1110100 —— 溢出

（5）A=115= 111 0011B, B= -24= -11 000B
[A]补=0 1110011, [B]补=1，110 1000
[A+B]补= 0 1110011 + 1 1101000 = 0 1011011——无溢出
A+B= 101 1011B = 91

26.按机器补码浮点运算步骤，计算[x±y]补.
（1）$x=2^{-011}\times$ 0.101 100，$y=2^{-010}\times$（-0.011 100）；
（2）$x=2^{-011}\times$（-0.100 010），$y=2^{-010}\times$（-0.011 111）；
（3）$x=2^{101}\times$（-0.100 101），$y=2^{100}\times$（-0.001 111）。

解：先将x、y转换成机器数形式：
(1) $x=2^{-011}\times$ 0.101 100，$y=2^{-010}\times$（-0.011 100）
[x]补=1，101；0.101 100, [y]补=1，110；1.100 100
[Ex]补=1,101, [y]补=1,110, [Mx]补=0.101 100, [My]补=1.100 100

1）对阶：

[ΔE]补=[Ex]补+[-Ey]补 = 11,101+ 00,010=11,111 < 0，

应 Ex 向 Ey 对齐，则：[Ex]补+1=11，101+00，001=11，110 = [Ey]补

[x]补=1，110；0.010 110

2）尾数运算：

[Mx]补+[My]补= 0.010 110 + 11.100 100=11.111010

[Mx]补+[-My]补=0.010 110 + 00.011100= 00.110 010

3）结果规格化：

[x+y]补=11，110；11.111 010 = 11，011；11.010 000 （尾数左规 3 次，阶码减 3）

[x-y]补=11，110；00.110 010, 已是规格化数。

4）舍入：无

5）溢出：无

则：$x+y=2^{-101}\times(-0.110\ 000)$

$x-y=2^{-010}\times 0.110\ 010$

(2) $x=2^{-011}\times(-0.100010)$，$y=2^{-010}\times(-0.011111)$

[x]补=1，101；1.011 110, [y]补=1，110；1.100 001

1） 对阶：过程同(1)的 1），则

[x]补=1，110；1.101 111

2）尾数运算：

[Mx]补+[My]补= 11.101111 + 11. 100001 = 11.010000

[Mx]补+[-My]补= 11.101111 + 00.011111 = 00.001110

3）结果规格化：

[x+y]补=11，110；11.010 000,已是规格化数

[x-y]补=11，110；00.001 110 =11，100；00.111000 （尾数左规 2 次，阶码减 2）

4）舍入：无

5）溢出：无

则：$x+y=2^{-010}\times(-0.110\ 000)$

$x-y=2^{-100}\times 0.111\ 000$

(3) $x=2^{101}\times(-0.100\ 101)$，$y=2^{100}\times(-0.001\ 111)$

[x]补=0，101；1.011 011, [y]补=0，100；1.110 001

1）对阶：

[ΔE]补=00，101+11，100=00，001 >0，应 Ey 向 Ex 对齐，则：

[Ey]补+1=00，100+00，001=00，101=[Ex]补

[y]补=0，101；1.111 000（1）

2）尾数运算：

[Mx]补+[My]补= 11.011011+ 11.111000（1）= 11.010011（1）

[Mx]补+[-My]补= 11.011011+ 00.000111（1）= 11.100010（1）

2） 结果规格化：

[x+y]补=00，101；11.010 011（1），已是规格化数

[x-y]补=00，101；11.100 010（1）=00，100；11.000 101 （尾数左规 1 次，

阶码减 1）

4）舍入：

[x+y]补=00，101；11.010 011（舍）

[x-y]补 不变

5）溢出：无

则：$x+y=2^{101}\times(-0.101\ 101)$

$x-y=2^{100}\times(-0.111\ 011)$

32. 设机器字长为 16 位，分别按 4、4、4、4 和 5、5、3、3 分组后，

（1）画出按两种分组方案的单重分组并行进位链框图，并比较哪种方案运算速度快。

（2）画出按两种分组方案的双重分组并行进位链框图，并对这两种方案进行比较。

（3）用 74181 和 74182 画出单重和双重分组的并行进位链框图。

解：（1）4—4—4—4 分组的 16 位单重分组并行进位链框图见教材 286 页图 6.22。

5—5—3—3 分组的 16 位单重分组并行进位链框图如下：

（2）4—4—4—4 分组的 16 位双重分组并行进位链框图见教材 289 页图 6.26。

5—5—3—3 分组的 16 位双重分组并行进位链框图如下：

5—5—3—3 分组的进位时间=2.5ty×3=7.5ty；

4—4—4—4 分组的进位时间=2.5ty×3=7.5ty；

可见，两种分组方案最长加法时间相同。

结论：双重分组并行进位的最长进位时间只与组数和级数有关，与组内位数无关。

（3）单重分组 16 位并行加法器逻辑图如下（正逻辑）：

注意： 1）74181 芯片正、负逻辑的引脚表示方法；

2）为强调可比性，5-5-3-3 分组时不考虑扇入影响；

3）181 芯片只有最高、最低两个进位输入/输出端，组内进位无引脚；

4）181 为 4 位片，无法 5-5-3-3 分组，只能 4-4-4-4 分组；

5）单重分组跳跃进位只用到 181，使用 182 的一定是双重以上分组跳跃进位；

6）单重分组跳跃进位是并行进位和串行进位技术的结合；双重分组跳跃进位是二级并行进位技术；特别注意在位数较少时，双重分组跳跃进位可以采用全先行进位技术实现；位数较多时，可采用双重分组跳跃进位和串行进位技术结合实现。

第 七 章

1. 什么叫机器指令？什么叫指令系统？为什么说指令系统与机器的主要功能以及与硬件结构之间存在着密切的关系？
答：参考 P300。

2. 什么叫寻址方式？为什么要学习寻址方式？
答：参看 P310。

3. 什么是指令字长、机器字长和存储字长？
答：略。

4. 零地址指令的操作数来自哪里？？各举一例说明。
答：零地址指令的操作数来自 ACC，为隐含约定。
在一地址指令中，另一个操作数的地址通常可采用 ACC 隐含寻址方式获得。

5. 对于二地址指令而言，操作数的物理地址可安排在什么地方？举例说明。
答：对于二地址指令而言，操作数的物理地址可安排在寄存器内、指令中或内存单元内等。

8. 某机指令字长 16 位，每个操作数的地址码为 6 位，设操作码长度固定，指令分为零地址、一地址和二地址三种格式。若零地址指令有 M 条，一地址指令有 N 种，则二地址指令最多有几种？若操作码位数可变，则二地址指令最多允许有几种？
解：1）若采用定长操作码时，二地址指令格式如下：

OP（4 位）	A1（6 位）	A2（6 位）

设二地址指令有 K 种，则：$K=2^4-M-N$
当 M=1（最小值），N=1（最小值）时，二地址指令最多有：Kmax=16-1-1=14 种
3） 若采用变长操作码时，二地址指令格式仍如 1）所示，但操作码长度可随地址码的个数而变。此时，$K= 2^4 -（N/2^6 + M/2^{12} ）$；
当（$N/2^6 + M/2^{12}$ ）≤1 时（$N/2^6 + M/2^{12}$ 向上取整），K 最大，则二地址指令最多有：Kmax=16-1=15 种（只留一种编码作扩展标志用。）

9. 试比较间接寻址和寄存器间接寻址。
答：略。

10. 试比较基址寻址和变址寻址。
略。

11. 画出先变址再间址及先间址再变址的寻址过程示意图。
解：1）先变址再间址寻址过程简单示意如下：
EA=[(IX)+A]，IX→(IX)+1
2）先间址再变址寻址过程简单示意如下：EA=(IX)+(A)，IX→ (IX)+1

16. 某机主存容量为 4M×16 位，且存储字长等于指令字长，若该机指令系统可完成 108 种操

作，操作码位数固定，且具有直接、间接、变址、基址、相对、立即等六种寻址方式，试回答：（1）画出一地址指令格式并指出各字段的作用；

（2）该指令直接寻址的最大范围；

（3）一次间址和多次间址的寻址范围；

（4）立即数的范围（十进制表示）；

（5）相对寻址的位移量（十进制表示）；

（6）上述六种寻址方式的指令哪一种执行时间最短？哪一种最长？为什么？哪一种便于程序浮动？哪一种最适合处理数组问题？

（7）如何修改指令格式，使指令的寻址范围可扩大到 4M？

（8）为使一条转移指令能转移到主存的任一位置，可采取什么措施？简要说明之。

解：（1）单字长一地址指令格式：

OP（7 位）	M（3 位）	A（6 位）

OP 为操作码字段，共 7 位，可反映 108 种操作；

M 为寻址方式字段，共 3 位，可反映 6 种寻址操作；

A 为地址码字段，共 16-7-3=6 位。

（2）直接寻址的最大范围为 2^6=64。

（3）由于存储字长为 16 位，故一次间址的寻址范围为 2^{16}；若多次间址，需用存储字的最高位来区别是否继续间接寻址，故寻址范围为 2^{15}。

（4）立即数的范围为-32——31（有符号数），或 0——63（无符号数）。

（5）相对寻址的位移量为-32——31。

（6）上述六种寻址方式中，因立即数由指令直接给出，故立即寻址的指令执行时间最短。间接寻址在指令的执行阶段要多次访存(一次间接寻址要两次访存，多次间接寻址要多次访存)，故执行时间最长。变址寻址由于变址寄存器的内容由用户给定，而且在程序的执行过程中允许用户修改，而其形式地址始终不变，故变址寻址的指令便于用户编制处理数组问题的程序。相对寻址操作数的有效地址只与当前指令地址相差一定的位移量，与直接寻址相比，更有利于程序浮动。

（7）**方案一**：为使指令寻址范围可扩大到 4M，需要有效地址 22 位，此时可将单字长一地址指令的格式改为双字长，如下图示：

OP（7 位）	MOD（3 位）	A（高 6 位）
A（低 16 位）		

方案二：如果仍采用单字长指令（16 位）格式，为使指令寻址范围扩大到 4M，可通过段寻址方案实现。安排如下：

硬件设段寄存器 DS（16 位），用来存放段地址。在完成指令寻址方式所规定的寻址操作后，得有效地址 EA（6 位），再由硬件自动完成段寻址，最后得 22 位物理地址。 即：物理地址=（DS）×2^6 + EA

注：段寻址方式由硬件隐含实现。在编程指定的寻址过程完成、EA 产生之后由硬件自动完成，对用户是透明的。

方案三：在采用单字长指令（16 位）格式时，还可通过页面寻址方案使指令寻址范围扩大到 4M。安排如下：

硬件设页面寄存器 PR（16 位），用来存放页面地址。指令寻址方式中增设页面寻址。当需要使指令寻址范围扩大到 4M 时，编程选择页面寻址方式，则：EA =（PR）‖A （有效地址=页面地址“拼接”6 位形式地址），这样得到 22 位有效地址。

（8）为使一条转移指令能转移到主存的任一位置，寻址范围须达到 4M，除了采用(7) 方

案一中的双字长一地址指令的格式外，还可配置 22 位的基址寄存器或 22 位的变址寄存器，使 EA = (BR) + A （BR 为 22 位的基址寄存器)或 EA =（IX）+ A(IX 为 22 位的变址寄存器)，便可访问 4M 存储空间。还可以通过 16 位的基址寄存器左移 6 位再和形式地址 A 相加，也可达到同样的效果。

总之，不论采取何种方式，最终得到的实际地址应是 22 位。

19. 某 CPU 内有 32 个 32 位的通用寄存器，设计一种能容纳 64 种操作的指令系统。假设指令字长等于机器字长，试回答以下问题：

（1）如果主存可直接或间接寻址，采用寄存器—存储器型指令，能直接寻址的最大存储空间是多少？画出指令格式并说明各字段的含义。

（2）在满足（1）的前提下，如果采用通用寄存器作基址寄存器，则上述寄存器—存储器型指令的指令格式有何特点？画出指令格式并指出这类指令可访问多大的存储空间？

解：（1）如采用 RS 型指令，则此指令一定是二地址以上的地址格式，指令格式如下：

OP（6 位）	R（5 位）	I(1 位)	A（20 位）

操作码字段 OP 占 6 位，因为 2^6>=64；

寄存器编号 R 占 5 位，因为 2^5>=32；

间址位 I 占 1 位，当 I=0，存储器寻址的操作数为直接寻址，当 I=1 时为间接寻址；

形式地址 A 占 20 位，可以直接寻址 2^{20}字。

（2）如采用基址寻址，则指令格式中应给出基址寄存器号，以指定哪一个通用寄存器用作基址寄存器。指令格式变为：

OP（6 位）	源 R（5 位）	I（1 位）	X（1 位）	目标 R（5 位）	A（14 位）

增加寻址特征位 X，当 X=1 时，以目标寄存器 R 作为基址寄存器进行基址寻址。

基址寻址可访问存储空间为：2^{32}字。

第八章

1. CPU 有哪些功能？画出其结构框图并简要说明各个部件的作用。
答：参考 P328 和图 8.2。

2. 什么是指令周期？指令周期是否有一个固定值？为什么？
解：指令周期是指取出并执行完一条指令所需的时间。

由于计算机中各种指令执行所需的时间差异很大，因此为了提高 CPU 运行效率，即使在同步控制的机器中，不同指令的指令周期长度都是不一致的，也就是说指令周期对于不同的指令来说不是一个固定值。

3. 画出指令周期的流程图，分析说明图中每个子周期的作用。
答：参看 P343 及图 8.8。

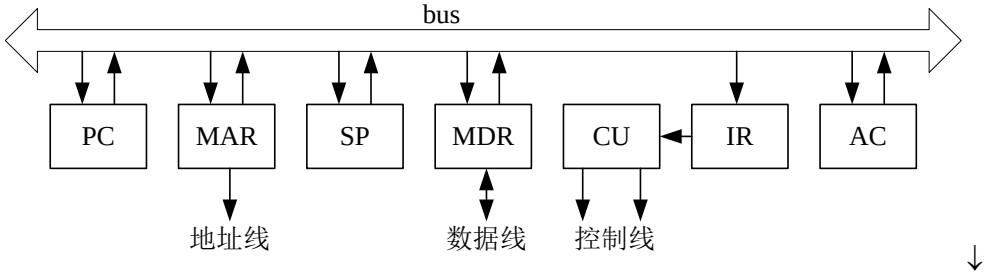
4. 设 CPU 内有下列部件：PC、IR、SP、AC、MAR、MDR 和 CU。

（1）画出完成间接寻址的取数指令 LDA@X（将主存某地址单元 X 的内容取至 AC 中）的数据流（从取指令开始）。

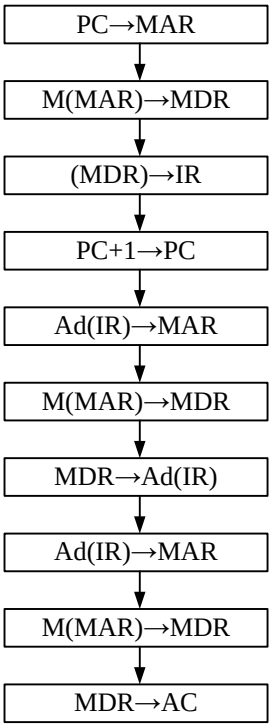
（2）画出中断周期的数据流。

解：CPU 中的数据流向与所采用的数据通路结构直接相关，不同的数据通路中的数据流是不一样的。常用的数据通路结构方式有直接连线、单总线、双总线、三总线等形式，目前大多采用总线结构，直接连线方式仅适用于结构特别简单的机器中。

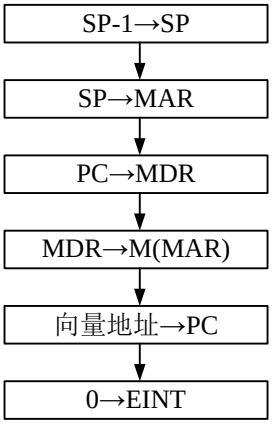
为简单起见，本题采用单总线将题中所给部件连接起来，框图如下：



（1）LDA@X 指令周期数据流程图：



（2）中断周期流程图如下：



注：解这道题有两个要素，首先要根据所给部件设计好数据通路，即确定信息流动的载体。其次选择好描述数据流的方法，无论采用什么样的表达方式，其关键都要能清楚地反映数据在通路上流动的顺序，即强调一个“流”字。较好的表达方式是流程图的形式。

5. 中断周期前是什么阶段？中断周期后又是什么阶段？在中断周期 CPU 应完成什么操作？
答：中断周期前是执行周期，中断周期后是取指周期。在中断周期，CPU 应完成保存断点、将中断向量送 PC 和关中断等工作。

7. 什么叫系统的并行性？粗粒度并行和细粒度并行有何区别？
答：所谓并行性包含同时性和并发性。同时性是指两个或两个以上的事件在同一时刻发生，并发性是指两个或多个事件在同一时间段发生。即在同一时刻或同一时间段内完成两个或两个以上性质相同或性质不同的功能，只要在时间上存在相互重叠，就存在并行性。

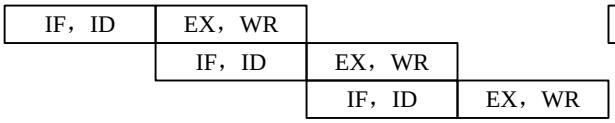
并行性又分为粗粒度并行和细粒度并行两类。粗粒度并行是指在多个处理机上分别运行多个进程，由多台处理机合作完成一个程序，一般用算法实现。细粒度并行是指在处理机的

指令级和操作级的并行性。

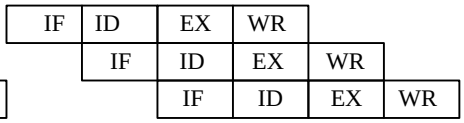
8. 什么是指令流水？画出指令二级流水和四级流水的示意图，它们中哪个更能提高处理机速度，为什么？

答：指令流水是指将一条指令的执行过程分为 n 个操作时间大致相等的阶段，每个阶段由一个独立的功能部件来完成，这样 n 个部件就可以同时执行 n 条指令的不同阶段，从而大大提高 CPU 的吞吐率。

指令二级流水和四级流水示意图如下：



二级指令流水示意图



四级指令流水示意图

四级流水更能提高处理机的速度。分析如下：

假设 IF、ID、EX、WR 每个阶段耗时为 t，则连续执行 n 条指令

采用二级流水线时，耗时为：4t+(n-1)2t=(2n+2)t

采用四级流水线时，耗时为：4t+(n-1)t=(n+3)t

在 n>1 时，n+3<2n+2，可见四级流水线耗时比二级流水线耗时短，因此更能提高处理机速度。

17. 在中断系统中 INTR、INT、EINT 三个触发器各有何作用？

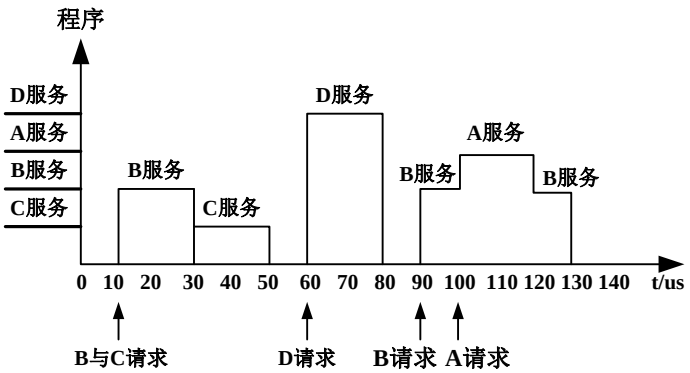
解：INTR——中断请求触发器，用来登记中断源发出的随机性中断请求信号，以便为 CPU 查询中断及中断排队判优线路提供稳定的中断请求信号。

EINT——中断允许触发器，CPU 中的中断总开关。当 EINT=1 时，表示允许中断（开中断），当 EINT=0 时，表示禁止中断（关中断）。其状态可由开、关中断等指令设置。

INT——中断标记触发器，控制器时序系统中周期状态分配电路的一部分，表示中断周期标记。当 INT=1 时，进入中断周期，执行中断隐指令的操作。

24. 现有 A、B、C、D 四个中断源，其优先级由高向低按 A、B、C、D 顺序排列。若中断服务程序的执行时间为 20μs，请根据下图所示时间轴给出的中断源请求中断的时刻，画出 CPU 执行程序的轨迹。

解：A、B、C、D 的响优先级即处理优先级。CPU 执行程序的轨迹图如下：



25. 某机有五个中断源 L0、L1、L2、 L3、L4，按中断响应的优先次序由高向低排序为 L0→

L1→L2→L3→L4，根据下示格式，现要求中断处理次序改为 L1→L4→L2→L0→L3，根据下面的格式，写出各中断源的屏蔽字。

解：各中断源屏蔽状态见下表：

中断源	屏蔽字				
	0	1	2	3	4
I0	1	0	0	1	0
I1	1	1	1	1	1
I2	1	0	1	1	0
I3	0	0	0	1	0
I4	1	0	1	1	1

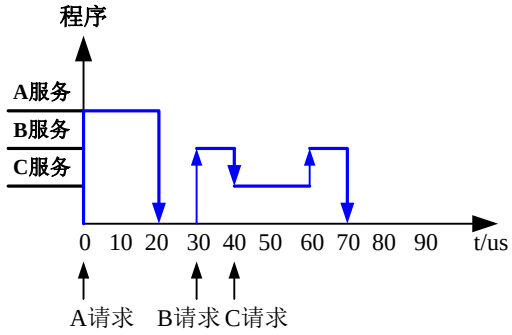
表中：设屏蔽位=1，表示屏蔽；屏蔽位=0，表示中断开放。

26. 设某机配有 A、B、C 三台设备，其优先顺序按 A→B→C 降序排列，为改变中断处理次序，它们的中断屏蔽字设置如下：

设备	屏蔽字
A	111
B	010
C	011

请按下图所示时间轴给出的设备请求中断的时刻，画出 CPU 执行程序的轨迹。设 A、B、C 中断服务程序的执行时间均为 20μ s。

解：A、B、C 设备的响应优先级为 A 最高、B 次之、C 最低，处理优先级为 A 最高、C 次之、B 最低。CPU 执行程序的轨迹图如下：



第九章

2. 控制单元的功能是什么？其输入受什么控制？
答：控制单元的主要功能是发出各种不同的控制信号。其输入受时钟信号、指令寄存器的操作码字段、标志和来自系统总线的控制信号的控制。

3. 什么是指令周期、机器周期和时钟周期？三者有何关系？
答：CPU 每取出并执行一条指令所需的全部时间叫指令周期；

机器周期是在同步控制的机器中，执行指令周期中一步相对完整的操作（指令步）所需时间，通常安排机器周期长度等于主存周期；

时钟周期是指计算机主时钟的周期时间，它是计算机运行时最基本的时序单位，对应完成一个微操作所需时间，通常时钟周期等于计算机主频的倒数。

4. 能不能说机器的主频越快，机器的速度就越快，为什么？
解：不能说机器的主频越快，机器的速度就越快。因为机器的速度不仅与主频有关，还与数据通路结构、时序分配方案、ALU 运算能力、指令功能强弱等多种因素有关，要看综合效果。

5. 设机器 A 的主频为 8MHz，机器周期含 4 个时钟周期，且该机的平均指令执行速度是 0.4MIPS，试求该机的平均指令周期和机器周期，每个指令周期中含几个机器周期？如果机器 B 的主频为 12MHz，且机器周期也含 4 个时钟周期，试问 B 机的平均指令执行速度为多少 MIPS?
解：先通过 A 机的平均指令执行速度求出其平均指令周期，再通过主频求出时钟周期，然后进一步求出机器周期。B 机参数的算法与 A 机类似。计算如下：

A 机平均指令周期=1/0.4MIPS=2.5μs
A 机时钟周期=1/8MHz=125ns
A 机机器周期=125ns×4=500ns=0.5μs
A 机每个指令周期中含机器周期个数=2.5μs÷0.5μs=5 个
B 机时钟周期 =1/12MHz≈83ns
B 机机器周期 =83ns×4=332ns
设 B 机每个指令周期也含 5 个机器周期，则：
B 机平均指令周期=332ns×5=1.66μs
B 机平均指令执行速度=1/1.66μs=0.6MIPS
结论：主频的提高有利于机器执行速度的提高。

6. 设某机主频为 8MHz，每个机器周期平均含 2 个时钟周期，每条指令平均有 4 个机器周期，试问该机的平均指令执行速度为多少 MIPS？若机器主频不变，但每个机器周期平均含 4 个时钟周期，每条指令平均有 4 个机器周期，则该机的平均指令执行速度又是多少 MIPS？由此可得出什么结论？
解：先通过主频求出时钟周期，再求出机器周期和平均指令周期，最后通过平均指令周期的倒数求出平均指令执行速度。计算如下：

时钟周期=1/8MHz=0.125×10^{-6}s
机器周期=0.125×10^{-6}s×2=0.25×10^{-6}s
平均指令周期=0.25×10^{-6}s×4=10^{-6}s

平均指令执行速度=1/10^{-6}s=1MIPS

当参数改变后：机器周期= 0.125×10^{-6}s×4=0.5×10^{-6}s

平均指令周期=0.5×10^{-6}s×4=2×10^{-6}s

平均指令执行速度=1/（2×10^{-6}s） =0.5MIPS

结论：两个主频相同的机器，执行速度不一定一样。

7. 某 CPU 的主频为 10MHz，若已知每个机器周期平均包含 4 个时钟周期，该机的平均指令执行速度为 1MIPS，试求该机的平均指令周期及每个指令周期含几个机器周期？若改用时钟周期为 0.4μs 的 CPU 芯片，则计算机的平均指令执行速度为多少 MIPS？若要得到平均每秒 80 万次的指令执行速度，则应采用主频为多少的 CPU 芯片？

解：先通过主频求出时钟周期时间，再进一步求出机器周期和平均指令周期。

时钟周期=1/10MHz=0.1×10^{-6}s

机器周期=0.1×10^{-6}s×4=0.4×10^{-6}s

平均指令周期=1/1MIPS=10^{-6}s

每个指令周期所含机器周期个数=10^{-6}s /0.4×10^{-6}s =2.5 个

当芯片改变后：机器周期=0.4μs×4=1.6μs

平均指令周期=1.6μs×2.5=4μs

平均指令执行速度=1/4μs=0.25MIPS

若要得到平均每秒 80 万次的指令执行速度，则:

平均指令周期=1/0.8MIPS=1.25×10^{-6}=1.25μs

机器周期=1.25μs÷2.5=0.5μs

时钟周期= 0.5μs÷4=0.125μs

CPU 主频=1/0.125μs=8MHz

8. 某计算机的主频为 6MHz，各类指令的平均执行时间和使用频度如下表所示，试计算该机的速度（单位用 MIPS 表示），若上述 CPU 芯片升级为 10MHz，则该机的速度又为多少？

指令类别	存取	加、减、比较、转移	乘除	其它
平均指令执行时间	0.6μs	0.8μs	10μs	1.4μs
使用频度	35%	45%	5%	15%

解: (1)指令平均运行时间 =（0.6×0.35+0.8×0.45+10×0.05+1.4×0.15） μs= 1.28μs

机器平均运行速度 = 1/1.28μs≈0.78MIPS

(2)时钟周期 = 1/6MHz≈0.167μs

指令平均运行周期数 = 1.28μs ÷0.167μs≈7.66CPI

若 CPU 芯片升级为 10MHz，时钟周期 = 1/10MHz=0.1μs

指令平均运行时间 = 0.1μs×7.66=0.766μs

机器平均运行速度 = 1/0.766μs≈1.3MIPS

10. 试比较同步控制、异步控制和联合控制的区别。

答：同步控制是指任何一条指令或指令中任何一个微操作的执行都是事先确定的，并且都受同一基准时标的时序信号所控制的方式。异步控制无基准时标信号，微操作的时序是由专门的应答线路控制，即控制单元发出执行某一微操作的控制信号后，等待执行部件完成了该操作后发回“回答”或“结束”信号，再开始新的微操作。联合控制是同步控制和异步控制相结合的方式，即大多数操作（如 CPU 内部各操作）在同步时序信号的控制下进行，少数时间难以确定的微操作（如涉及 I/O 操作）采用异步控制。

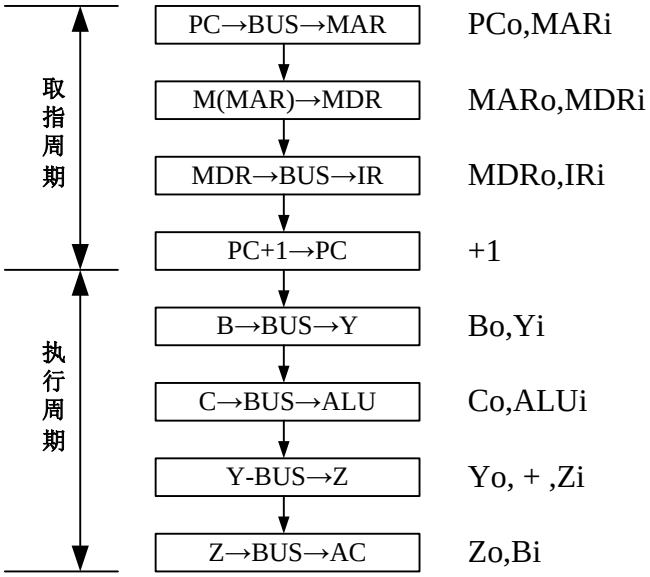
11. 设 CPU 内部结构如图 9.4 所示，此外还设有 B、C、D、E、H、L 六个寄存器，它们各自的输入和输出端都与内部总线相通，并分别受控制信号控制（如 Bi 为寄存器 B 的输入控制；Bo 为 B 的输出控制）。要求从取指令开始，写出完成下列指令所需的全部微操作和控制信号。

（1）ADD B，C　　((B)+(C) →B)

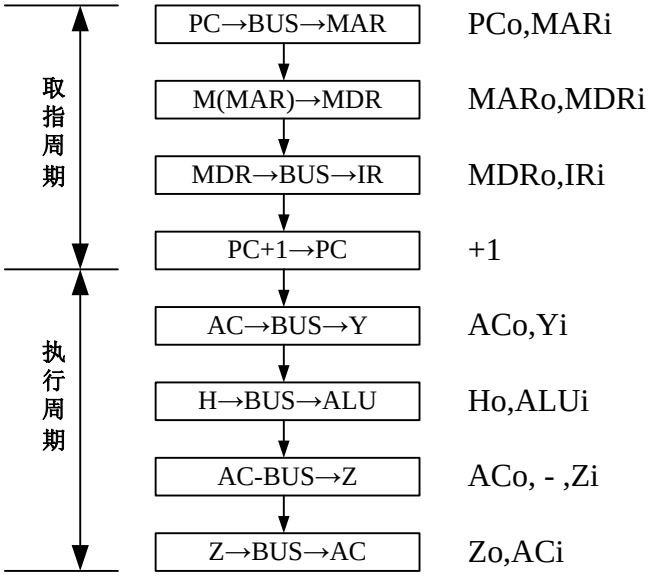
（2）SUB A,H　　((AC)-(H) →AC)

解：先画出相应指令的流程图，然后将图中每一步数据通路操作分解成相应的微操作，再写出同名的微命令即可。

（1） ADD B，C 指令流程及微命令序列如下：



（2） SUB A，H 指令流程及微命令序列如下：



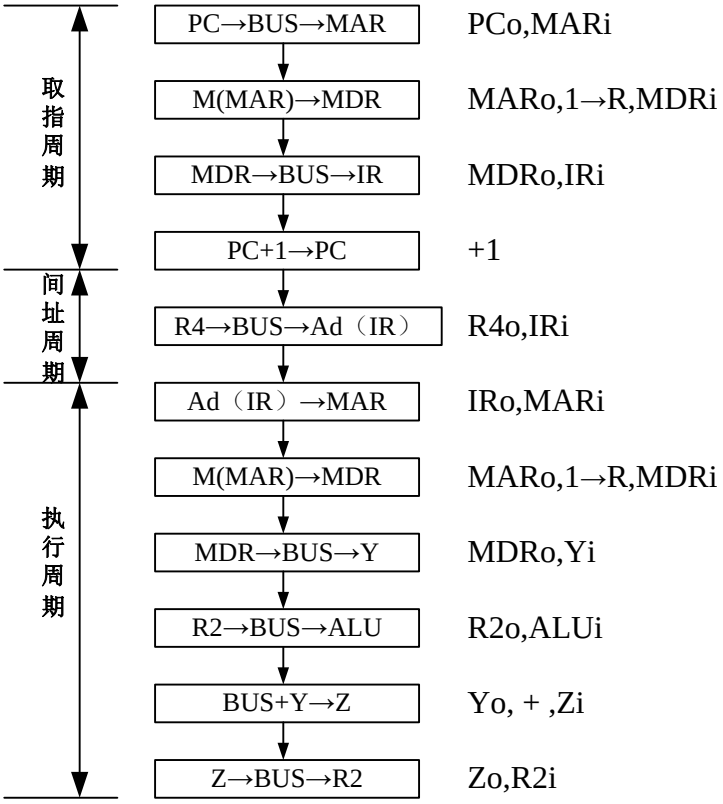
13. 设 CPU 内部结构如图 9.4 所示，此外还设有 R1~R4 四个寄存器，它们各自的输入和输出端都与内部总线相通，并分别受控制信号控制（如 R2i 为寄存器 R2 的输入控制；R2o 为 R2

的输出控制)。要求从取指令开始，写出完成下列指令所需的全部微操作和控制信号。

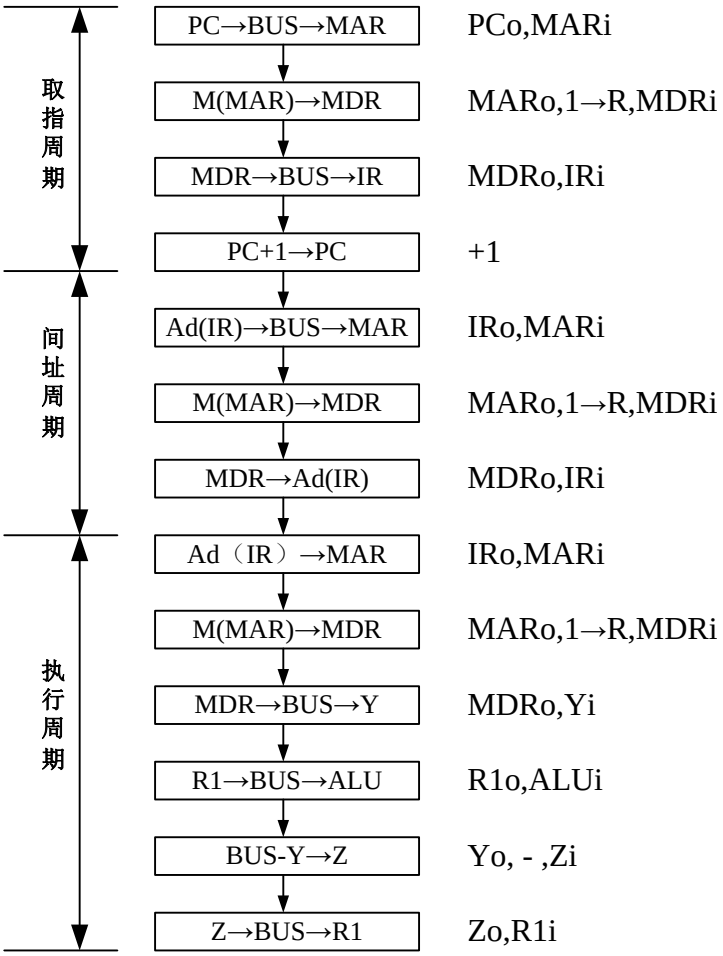
（1）ADD R2，@R4 ； ((R2)+((R4)) →R2，寄存器间接寻址)

（2）SUB R1,@mem ； ((R1)-((mem)) →R1，存储器间接寻址)

解：（1）ADD R2，@R4 的指令周期信息流程图及微操作控制信号如下：



（2）SUB R1,@mem 指令周期信息流程图及微操作控制信号如下：



14. 设单总线计算机结构如图 9.5 所示，其中 M 为主存，XR 为变址寄存器，EAR 为有效地址寄存器，LATCH 为锁存器。假设指令地址已存于 PC 中，画出“LDA *D”和“SUB X,D”指令周期信息流程图，并列出相应的控制信号序列。

说明：

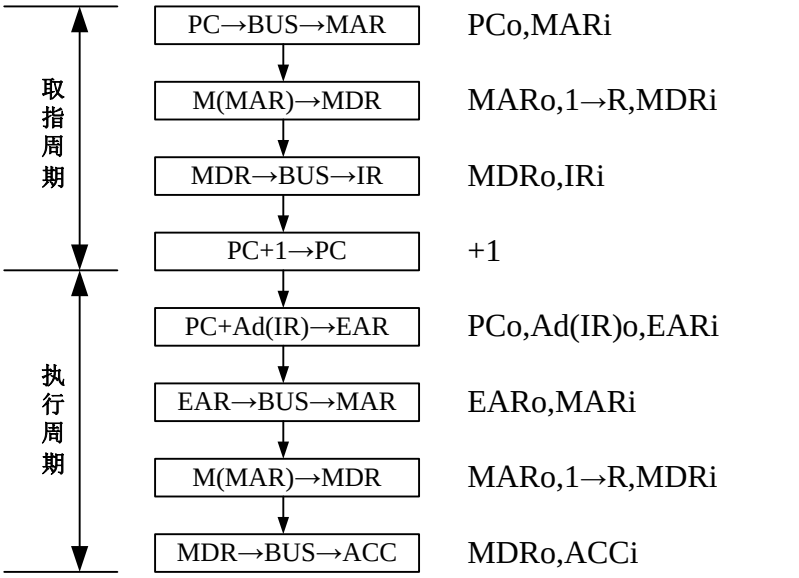
（1）“LDA *D”指令字中*表示相对寻址，D 为相对位移量。

（2）“SUB X,D”指令字中 X 为变址寄存器 XR，D 为形式地址。

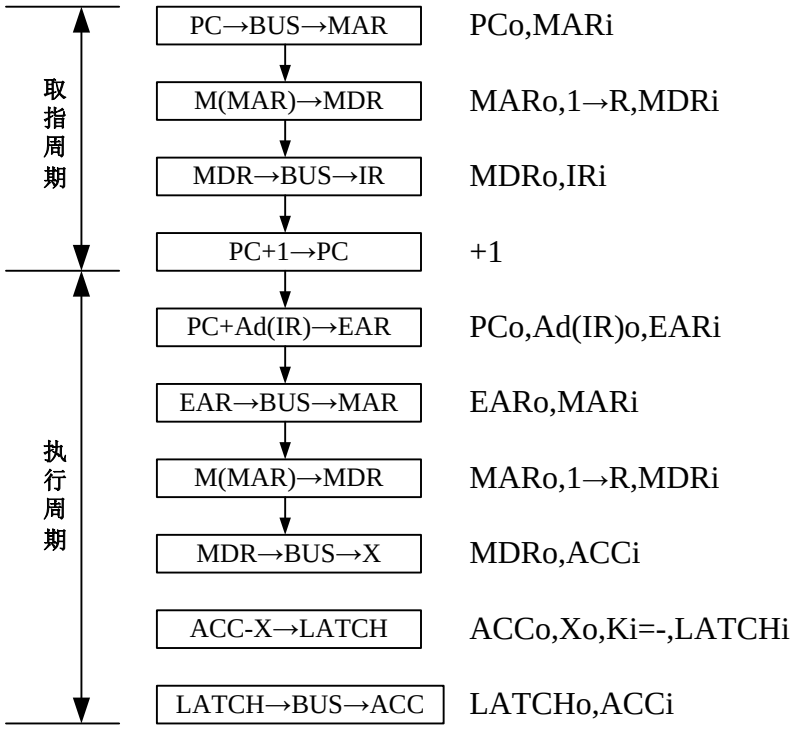
（3）寄存器的输入和输出均受控制信号控制，例如，PCi 表示 PC 的输入控制信号，MDRo 表示 MDR 的输出控制信号。

（4）凡是需要经过总线实现寄存器之间的传送，需要在流程图中注明，如 PC→Bus→MAR，相应控制信号为 PCo 和 MARi。

解：（1）“LDA *D”指令周期信息流程图及微操作控制信号如下：



（2）“SUB X,D” 指令周期信息流程图及微操作控制信号如下：



第十章

1. 假设响应中断时，要求将程序断点存在堆栈内，并且采用软件办法寻找中断服务程序的入口地址，试写出中断隐指令的微操作及节拍安排。

解：设软件查询程序首址为 0 号内存单元，则中断隐指令的微操作命令及节拍安排如下：

T0　　0 SP

T1　　，SP→ MAR

T2　　SP→W，SP+1 M（MAR）

T3　　PSW→MAR，MDR→SP

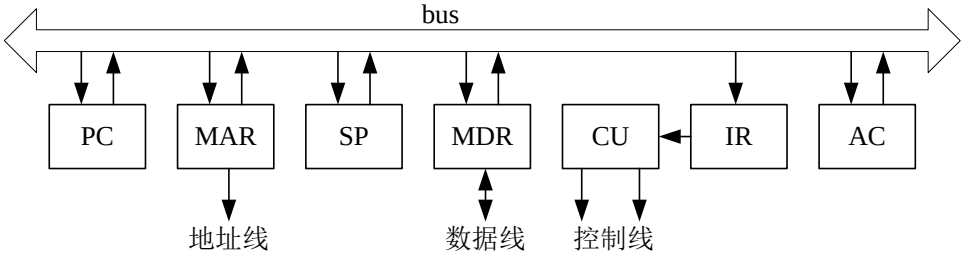
T4　　1→ W，SP+1→MDR，　→M（MAR） PC→EINT　　由于题意中没有给出确切的数据通路结构，故上述节拍分配方案的并行性较低。→PC，MDR→0　　→MDR，1→MAR

2. 写出完成下列指令的微操作及节拍安排（包括取指操作）。

（1）指令 ADD R1，X 完成将 R1 寄存器的内容和主存 X 单元的内容相加，结果存于 R1 的操作。

（2）指令 ISZ X 完成将主存 X 单元的内容增 1，并根据其结果若为 0，则跳过下一条指令执行。

解：设采用单总线结构的 CPU 数据通路如下图所示，且 ALU 输入端设两个暂存器 C、D（见 17 题图）。并设采用同步控制，每周期 3 节拍：



（1）指令 ADD R1，X 的微操作及节拍安排如下：

取指周期：T0　PC→MAR，1→ R

T1　M(MAR) →MDR，PC+1→PC

T2　MDR→IR,OP(IR) →ID

执行周期 1：T0 Ad（IR）→MAR，1→R

T1　M(MAR) →MDR

T2　MDR→D

执行周期 2：T0 R1→C

T1　+

T2　ALU→R1

（2）指令 ISZ X 的微操作及节拍安排:

取指周期同（1）:略

执行周期 1:　T0　Ad（IR）→MAR，1→R

T1　M(MAR)→MDR

T2　MDR→C，+1→ALU

执行周期 2：T0　ALU→MDR，1→W

T1　(PC+1)・Z+ PC・$\overline{Z}$ →PC

12. 能否说水平型微指令就是直接编码的微指令，为什么？

解：不能说水平型微指令就是直接编码的微指令，因为符合水平型微指令特征的微指令都属于水平型微指令，常见的有：直接编码、字段直接编码、字段间接编码，及混合编码等。直接编码的微指令只是最典型的一种。

15. 设控制存储器的容量为 512×48 位，微程序可在整个控存空间实现转移，而控制微程序转移的条件共有 4 个（采用直接控制），微指令格式如下：

解：因为控制存储器共有 $512*48=2^9*48$

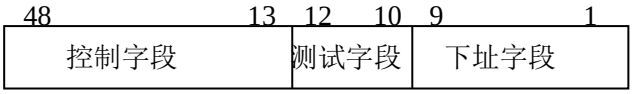
所以，下址字段应有 9 位，微指令字长 48 位

又因为控制微程序转移的条件有 4 个，$4+1<=2^3$

所以判断测试字段占 3 位

因此控制字段位数为：48-9-3=36

微指令格式为：

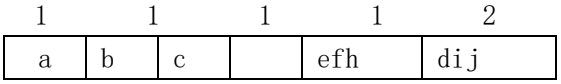


21. 下表给出 8 条微指令 I1~I8 及所包含的微命令控制信号，设计微指令操作控制字段格式，要求所使用的控制位最少，而且保持微指令本身内在的并行性。

解：为使设计出的微指令操作控制字段最短，并且保持微指令本身内在的并行性，应采用混合编码法。首先找出互斥的微命令组，为便于分析，将微命令表重画如下：

由表中微命令的分布情况可看出：a、b、c、d、e 微命令的并行性太高，因此不能放在同一字段中。另外，由分析可知，在 2、3、4 分组的互斥组中，3 个一组的微命令互斥组对控制位的压缩作用最明显。因此，应尽可能多的找出 3 个一组的互斥组。现找出的互斥组有：cfj，dij，efh，fhi，bgj，ehj，efj……等等。

从中找出互不相重的互斥组有两个：dij，efh。则：微指令操作控制字段格式安排如下：



各字段编码分配如下： a：0 无操作，1 a 微命令；

b：0 无操作，1 b 微命令；

c：0 无操作， 1 c 微命令；

g ：0 无操作，1 g 微命令；

dij ：00 无操作；01 d 微命令；10 i 微命令；11 j 微命令；

efh：00 无操作；01 e 微命令；10 f 微命令；11 h 微命令

与采用直接控制法比较：直接控制法：10 个微命令需 10 位操作控制位；本方案中 10 个微命令需 8 位操作控制位，压缩了 2 位。