简答题

区别与联系、相同点和不同点

1. 存储系统层次结构 P125

(1) Cache-主存存储层次（Cache 存储系统）

① 为解决主存速度不足提出。在 Cache 和主存之间，增加辅助硬件，让他们构成一个整体。

② 从 CPU 看，速度接近 Cache 的速度，但有着主存的容量和接近主存的每位价格。

③ 由硬件调度，对系统程序员和应用程序员都透明。

(2) 主存-辅存存储层次（虚拟存储系统）

① 为解决主存容量不足提出。在主存和辅存之间，增加辅助的软硬件，让他们构成一个整体。

② 从 CPU 看，速度接近主存的速度，容量是虚拟的地址空间，但每位价格接近辅存。

③ 由操作系统调度，仅对应用程序员透明。

答:相同点：
1）出发点相同，二者的出发点都是为了提高存储系统性能价格比，都是力图使性能接近高速存储器，而价格接近低速存储器；
2）原理相同，都利用了程序运行局部性原理。
不同点：
1）目的不同，cache-主存解决速度差异，而主存－辅存解决主存容量不足问题
2）数据通路不同：CPU 与 cache 和主存皆有直接访问数据通路，而辅存则没有与 CPU 直接相通的数据通路
3）透明性不同：cache 管理完全由硬件完成，对系统程序和应用程序均透明，而虚存管理由软件（操作系统）和硬件共同完成，对系统程序不透明
4）未命中时的损失不同：由于 cache、主存、辅存存取时间差别不一，因此虚存未命中时的损失将远大于 cache 未命中时的损失。 +10 分

2. Cache 的读写操作 P159

(1) 读操作

① 如果 Cache 命中，则直接对 Cache 进行读操作，与主存无关。

② 如果 Cache 不命中，需要访问主存，并把该块信息一次从主存调入 Cache 内，如果 Cache 已满，就必须根据替换算法，替换掉原来的某块信息。

(2) 写操作

① 如果 Cache 命中，可能会遇到 Cache 与主存中内容不一致的问题，所以需要进行写处理。

② 写回法：只把数据写入 Cache，需要替换时才将已修改的 Cache 写回主存；写直达法：既写入 Cache 又写入主存。

③ 如果 Cache 不命中，就直接把信息写入主存。

④ 不按写分配法：只把信息写入主存；不按写分配法：写入主存后还要把该块从内存读入 Cache 中。

(3) 全相联映像

① 主存中的任何一块均可以装入 Cache 中的任何一块

(4) 直接映像

① 主存中每一个块只能被放置到 Cache 中唯一的一个指定位置，若产生块冲突，则原来的块将无条件地被换出。

(5) 组相联映像

① 将 Cache 空间分成大小相同的组，组间采取直接映像，组

内采取全相联映像。

3. CPU 控制方式 P181

(1) 同步控制方式

① 固定时序的控制方式，各项操作都由统一的时序信号控制，在每个机器周期中产生统一数目的工作脉冲。容易造成时间浪费。

(2) 异步控制方式

① 可变时序的控制方式，各项操作不采用统一的时序信号控制，而是根据指令的具体情况决定，需要多少时间就占用多少时间。各操作之间的衔接由“结束-起始”信号来实现。控制较复杂。

(3) 联合控制方式

① 同步和异步方式结合，在功能部件内部采用同步方式，在功能部件之间采用异步方式。

4. 流水线技术 P202

(1) 将一个较复杂的处理过程分成多个复杂程度相当、处理时间大致相等的子过程，每个子过程由一个独立的功能部件来完成、并行执行，处理对象在各子过程连成的线路上连续流动。

(2) 处理级别分类、单多功能分类、静态动态工作方式分类、线性非线性结构分类。

(3) 影响流水线性能的因素

① 结构相关：多条指令在同一时刻争夺同一资源形成冲突。

② 数据相关：后续指令要使用前面指令的操作结果，但这一结果未产生或未送到相应的位置。

③ 控制相关：遇到转移指令，无法确定接下来应该安排执行哪一程序段。

5. 集中式总线仲裁方式 P221

(1) 链式查询方式

① 优先次序由 BG 线上串接部件的先后位置确定，在查询链中离总线控制器最近的设备具有最高优先权。

② 优点：容易实现和扩充

③ 缺点：对查询链的故障敏感

(2) 计时器定时查询方式

① 计数器计数，定时地查询各个部件以确定是谁发出的请求。如果计数从 0 开始，优先级和练市查询方式相同；如果从中止点开始，则为循环优先级，各个部件使用总线的机会相等。

(3) 独立请求方式

① 每一个共享总线的部件均有一对控制线：总线请求 BR 和总线批准 BG。当某个部件请求使用总线时，便发出 BR，总线控制器中有排队电路，根据优先次序决定首先响应哪个部件，然后给它送回批准信号。

② 优点：响应时间快，便于改变优先级次序

③ 缺点：控制线数目多

6. 输入输出信息传送控制方式 P291

(1) 程序查询方式

① 输入输出完全通过 CPU 执行程序来完成的，CPU 全程参与，若外设未就绪则踏步等待。

(2) 程序中断方式

① CPU 启动外设后，无需等待，而是继续执行原来的程序；外设做好输入输出准备后，再向 CPU 发送中断请求，CPU 才转向执行中断服务程序，中断处理完毕后再返回原来的任务的断点处。

(3) 直接存储器存取（DMA）方式

① 在主存和外设之间开辟数据通路，DMA 控制器从 CPU 处完全接管总线的控制，进行以数据块为单位的传送。数据交换不经过 CPU，只需要在传送的起始和结束时由 CPU 处理。

(4) I/0 通道控制方式

① CPU 启动通道，由通道执行通道程序，完成输入输出操作。通道是从属于 CPU 的专用处理器。

⑴中断方式是程序切换，需要保护和恢复现场；DMA 方式除了开始和结束，不占用 CPU 的任何资源
⑵中断请求的响应时间只能发生在每条指令执行完毕的公操作阶段；DMA 请求的响应时间可在每个机器周期结束时
⑶中断传送过程完全依赖 CPU 执行指令实现；DMA 传送过程不需要 CPU

的干预，DMA 控制器控制数据传送过程，数据传输速率非常高，适合于高速外设的成组数据传送
⑷DMA 请求的优先级高于中断请求
⑸中断方式具有对异常事件的处理能力，DMA 方式仅局限于完成传送数据块的 I/O 操作

7. 中断和子程序调用的区别 P68 P296

(1) 程序中断是指计算机执行现行程序的过程中，出现某些异常情况和特殊请求，CPU 暂时中止现行程序，而转去对随机发生的更紧迫事件进行处理，处理完毕后自动返回原来程序继续执行。

(2) 子程序是由程序员事先安排好的，而中断服务程序的执行是由随机的中断事件引起的。

(3) 子程序的执行受到主程序或上层子程序的控制，而中断服务程序与被中断的现行程序无关。

(4) 不存在同时调用多个子程序的情况，但可能发生多个外设同时请求 CPU 服务，产生多个中断请求的情况。

(5) 因此，中断的处理要比调用子程序指令的执行复杂很多。

了解

1. 变型或组合寻址方式 P61

2. 堆栈结构 P63

(1) 寄存器堆栈（硬堆栈）

① 使用相互连接的寄存器组实现，之间具有自动推移的功能。栈顶位置固定。

(2) 存储器堆栈（软堆栈）

① 主存划出区域堆栈，栈底位置固定，栈顶位置浮动，由 SP 记录。

② 自底向上生成（向低地址方向）：进栈时先 SP-1

③ 自顶向下生成（向高地址方向）：进栈时先 SP+1

3. 虚拟存储器 P164

(1) 页式虚拟存储器

(2) 段式虚拟存储器

4. 周期 P178

(1) 指令周期

(2) 机器周期

(3) 时钟周期

5. 端口地址编址方式 P70 P289

(1) I/0 映射方式（独立编址）

(2) 存储器映射方式（统一编址）

计算题

1. 定点数补码、变形补码 加减法计算

(1) 码制转换 P20

正数：原码 = 反码 = 补码

负数：原码—$\xrightarrow{\text{取反}}$→反码—$\xrightarrow{+1}$→补码—$\xrightarrow{\text{符号位取反}}$→移码

“N 位补码”是包括符号位的说法。

原码和反码表示的正负数范围相对于 0 而言是对称的，它们都有 +0 和 -0 之分。补码的 +0 和 -0 表示相同，都是 0 后全零；1 后全零被用来表示最负的数 -2^{n-1} 或 -1 。

(2) 定点数补码加减法 P86

$[X+Y]_{补} = [X]_{补} + [Y]_{补}$　　$[X-Y]_{补} = [X]_{补} + [-Y]_{补}$

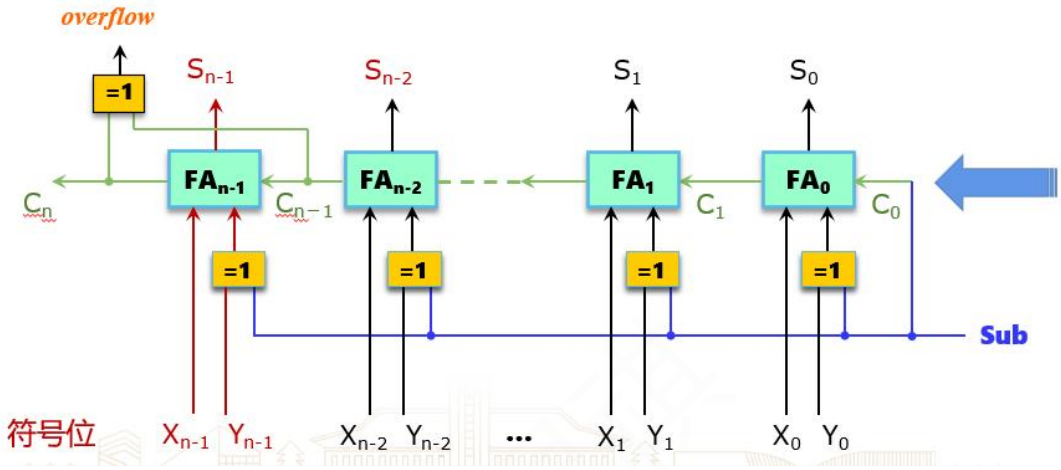
$[-Y]_{补}$是$[Y]_{补}$的机器负数，是将$[Y]_{补}$连同符号位一起取反，再+1 得到。也可以先对真值取相反数，然后再转换得到补码。

符号位参与竖式运算。最后得到的补码要转换回真值。

(3) 溢出判断 P87 P81

到做竖式加法运算的一步，若两数异号则不会溢出；若两数同号，而结果异号，则发生了正溢/负溢。

溢出 OF = $\underline{X_sY_sS_s} + X_sY_s\underline{S_s}$ = $C_s \oplus C_1$ = $S_{s1} \oplus S_{s2}$



(4) 变形补码 P88

变形补码是在补码的基础上多加一位符号位，其值和原来的符号位相同。当运算结果的两个符号位不同时，则发生了溢出。01 正溢，10 负溢。仅在运算时再扩充为双符号位。

2. 磁盘参数计算 P244

(1) 有效存储区域径向长度 = (外径 + 内径) / 2

(2) 面磁道数 = 有效径向长度 × 道密度

(3) 磁道长度(以最内圈磁道代替) = π × 内径

(4) 道容量 = 磁道长度 × 位密度

(5) 总存储容量 = 记录面数量 × 磁道数 × 每道容量

(6) 数据传输率 D_r = 转速 × 道容量

(7) 平均存取时间 T_a = 平均寻道时间 + 平均等待时间

$$= \frac{0+(磁道数-1)\times 道间移动时间}{2}+\frac{0+旋转一圈的时间}{2}$$

旋转一圈的时间 = 1 ÷ 转速

3. 字符显示器参数计算 P275

(1) 缓存容量 = 列 × 行 × 单个字符编码所占空间，VRAM 缓存默认存放字符的 ASCII 码，每个占 1 字节

(2) 字符发生器容量 = 字库字符数 × 点阵，ROM

(3) 字符窗口大小 = (点阵列+字间隔) × (点阵行+排间隔)

(4) 行列数和 VRAM 地址单元均从 0 开始计数，addr=列*X+Y

(5) 分频

① 点计数器：点阵列 + 字间隔

② 字计数器：屏幕列 + 水平回扫折合字符数

③ 行计数器：点阵行 + 排间隔

④ 排计数器：屏幕行 + 垂直回扫折合字符排数(默认 1)

(6) 主振频率=点频 = 场频×排分频×行分频×字分频×点分频

(7) 一个点计数循环访问一次 VRAM，一个字计数循环发送一次水平同步信号，一个排计数循环发送一次垂直同步信号

访问缓存时间间隔 = 1÷(场频×排分频×行分频×字分频)

(8) 屏幕上每个字符位置对应 VRAM 中的一个单元，各单元地址随着屏幕位置根据从左至右、从上至下的显示顺序，从低地址向

高地址安排。

4. 图形显示器参数计算 P277

(1) 缓存 VRAM 大小 = 列 × 行 × $\log_2$ 颜色数

(2) 每个位平面容量 = 行 × 列

(3) 点计数器：8 分频，每发送一次字节脉冲访问一次 VRAM

(4) 列(字节)计数器：分辨率列÷8 + 行线逆程回扫折合点数

(5) 行计数器：分辨率行 + 场逆程回扫折合列数

(6) VRAM 地址 = 行号×分辨率列÷8 + 列号

5. 总线带宽计算 P221

(1) 系统最大 I/O 传送速率 = n × 选择通道传送速率 + m × 多路通道传送速率

(2) 选择通道传送速率对各设备取 max，多路则求和

(3) 总线宽度指总线的线数，地址线宽度决定了能直接访问的地址空间范围，数据线宽度决定了一次访问能交换的数据位数。

(4) 总线带宽为最大数据传输率 B = W × F ÷ N

(5) W 为数据总线宽度，F 为总线时钟频率，N 为一次数据传送所需时钟周期数。N 默认取 1。

(6) 数据块字长为数据总线宽度。

设计分析题

1. 芯片刷新、字位扩展与片选 P133、P141

(1) 横向位扩展，纵向字扩展。

(2) 读写控制 WE 连到每一片。

(3) 地址线 A 的高位连入译码器后产生片选信号 CS，CS 分成字数个，每个 CS 连一行芯片，片选时选中整行（全字选中）。

(4) 地址线 A 的低位连到每一片，用做片内字选地址。

(5) 数据线 D 按芯片位数分组，每组连一列芯片。

(6) 部分译码法可能出现地址重叠，即一个存储单元出现多个地址。

(7) 刷新周期 = 读写周期，最大刷新间隔默认 2ms

(8) 存储器刷新时对所有芯片同时刷新，A×B 芯片刷新一遍所需刷新操作次数 = 存储矩阵行数 = $\sqrt{A}$ （单位：bit）

(9) 集中刷新：死区 = $\sqrt{A}$ × 刷新周期

(10) 分散刷新：每一个读写周期后紧跟一个刷新周期

(11) 异步刷新：相邻两行的刷新间隔 = 最大刷新间隔 ÷ 行数

2. 指令格式、寻址方式 P56

(1) 指令格式：操作码、寻址方式、地址(寄存器地址或主存地址)、立即数。

(2) 基本概念：形式地址 A、有效地址 EA

(EA/R_i) 代表地址 EA 或寄存器 R_i 中存放的数值。

① 立即寻址：操作数 S = 立即数 imm

② 寄存器寻址：S = (R_i)

③ 直接寻址：S = (A)，EA = A

④ 间接寻址：S = ((A))，可多级间址，@=1 时代表仍需间址

⑤ 寄存器间址：S = ((R_i))，EA = (R_i)

⑥ 变址寻址：S = ((R_x) + A)，加法要送 ALU 进行

EA = (R_x) + A，寻址范围为 $2^{Rx\ 的位数}$ 个存储单元

⑦ 基址寻址：S = ((R_b) + D)

⑧ 相对寻址：S = ((PC) + D)，寻址范围为 2^D 个存储单元

(PC) - 2^{n-1} ~ (PC) + 2^{n-1} - 1

⑨ 页面寻址：页面号(0、$(PC)_H$、页寄存器) // 页内地址 A

⑩ 前变址：先变址后间址(EA)；后变址：先间址(A)后变址

(3) 计算访存次数时，需要包括取指令与结果写回阶段，注意指令字长、数据字长与存取宽度的关系。

3. 数据通路和微操作指令 P183 P196

(1) 取指周期

1	(PC)→MAR	PC_{out} , MAR_{in}
2		Read
3	M(MAR)→MDR	
4	(MDR)→IR	MDR_{out} , IR_{in}
5	(PC)+1→PC	PC+1

(2) 执行周期（间址周期/取数周期、执行周期）

(3) 加法运算：(R0) + (A) → AC，$R0_{out}$, ADD （ALU 相当于选择器）

(4) 指令译码：OP(IR)→ID； 取 IR 中地址数：Ad(IR)

(5) 结果为零转移： $Z \cdot (MDR) + \overline{Z} \cdot (PC) \rightarrow PC$

(6) 寄存器：MAR, MDR, PC, IR

4. 中断

(1) 中断屏蔽码：为 1 的可以屏蔽，一般屏蔽同级

(2) CPU（现行程序）也可以有屏蔽码

(3) 中断请求到来时，若当前为现行程序，先按升级前响应次序短暂进入中断处理，再考虑由中断升级导致的抢占去向。

(4) 中断嵌套结束时，执行完当前等级的中断处理程序后，先短暂返回上一层，再考虑由屏蔽码导致的抢占去向。

表 8-3 中断屏蔽码

程序级别	屏蔽码				
	0 级	1 级	2 级	3 级	4 级
第 0 级	1	0	1	0	1
第 1 级	1	1	1	1	1
第 2 级	0	0	1	0	0
第 3 级	1	0	1	1	1
第 4 级	0	0	1	0	1

(2) 5 级中断同时发出中断请求，各级中断处理过程示意如图 8-18 所示。

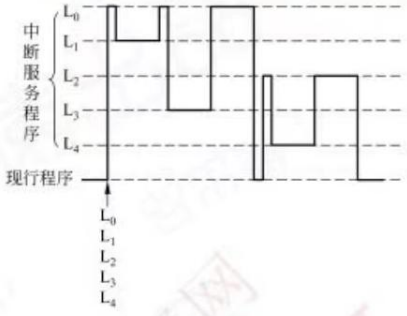


图 8-18 中断处理过程示意

简答题备选

6. 存储器设计思想 P8

(1) 冯诺依曼架构

(2) 哈佛架构

7. 指令系统发展方向 P74 P208

(1) 复杂指令系统（CISC）

(2) 精简指令系统（RISC）

8. 地址安排方案 P127

(1) 大端方案

(2) 小端方案

9. 动态 RAM 的刷新方式 P133

(1) 集中刷新方式

(2) 分散刷新方式

(3) 异步刷新方式

10. 片选信号的产生 P143

(1) 线选法

(2) 全译码法

(3) 部分译码法

11. Cache 更新策略 P161

(1) 写直达法

(2) 写回法