

## 书面作业二

刘泓尊 2018011446 计 84

### 1. ×

虚拟存储器的目的是给所有程序提供相同的虚拟地址空间，方便操作系统、链接器和装载器的设计。同时给程序员提供比主存大得多的地址空间。

但是，虚拟存储器的存在并不能加快外存访问速度，因为无论是段式存储还是页式存储，都需要在主存中存储段表或页表，多次访存之后获得物理地址，进而到外存中去访问，这一过程甚至增加了外存访问速度。

### 2. ×

高速缓存是为了提高 CPU 对存储器的平均访问速度。并不能提高主存的容量。

### 3. ×

RAID4 在 RAID5 之上进行优化，不再将校验放在一块磁盘，而是分散于每块磁盘，避免了校验盘的瓶颈问题。两者都只能坏 1 块磁盘。RAID5 并不能带来校验能力的提升。

### 4. D

ROM 为 8bit, 所以需要  $4KB / 2KB = 2$  片。

RAM 为 4bit, 所以需要  $2 \times 60KB / 4KB = 30$  片。

### 5. B

RISC-V 中，异常发生时，处理器将当前 PC 保存到 `mepc`，将异常原因保存到 `mcause`。操作系统完成异常处理后，将 PC 恢复为 `mepc`。但是通用寄存器的保存是由操作系统软件维护的，硬件只提供对应的读写指令。

### 6. C

全相联映射使得物理地址可以被映射到 cache 的每一行，而直接映射的一个物理地址只能被映射到 cache 的某一行，组相联作为两者的折中，将一个物理地址映射到多个 cache line。全相联映射选择 cache line 时灵活性更大，所以缓存命中率不小于另外两种方式。

### 7. C

THINPAD 教学计算机使用 2 组 SRAM 作为内存，SRAM 不需要定时刷新。DRAM 才需要定时刷新。

### 8. C

缓存缺失包括必然缺失、容量缺失、冲突缺失和无效缺失。而页面缺失与页表相关，会触发缺页异常。

### 9. C

TLB 缺失后，触发 TLB miss exception，处理器会进一步访问页表，得到物理地址。如果找不到还会触发缺页异常，由操作系统从硬盘对应页面换入内存。所

以 TLB 缺失不会导致程序出错。

#### 10. C

固态硬盘的物理地址包括 Die, Plane, Block 等, 由位于文件系统和物理介质之间的 FTL(闪存转换层)完成逻辑地址到物理地址的转换。使得操作系统可以按照传统的扇区方式访问固态硬盘。当文件系统发送指令说要写入或者更新一个特定的逻辑扇区时, FTL 实际上写入了另一个空闲物理页, 并更新映射表, 再把这个页上包含的旧数据标记为无效。

#### 11. B

静态存储器不需要刷新, 掉电之后需要重新写入数据, 与磁介质不同。动态存储器的读操作才是破坏性的, 会强制清零, 读出之后需要将数据写回; 但静态存储器不需要。固态硬盘采用闪存(Flash)或 DRAM 作为存储介质, 而不是静态存储器。

#### 12. 容量缺失、冲突缺失

#### 13. 2TB, 1.25TB

解: RAID1 方式采用 1 块存储盘和 1 块备份盘, 空间利用率 50%。RAID5(4+1)使用 4 份空间做存储, 1 份空间做校验, 空间利用率 80%。

#### 14. 91%, 27.2ns

Cache 命中率:  $1 - 450 / 5000 = 91\%$

平均访问时间:  $91\% \times 20 + 9\% \times 100 = 27.2\text{ns}$

#### 15.

##### (1).

页面大小  $64\text{B} = 2^6 \text{B}$ , 所以 VPO, PPO 位宽 6bit。

因为 VA14bit, PA12bit, 所以 VPN 8bit, PPN 6bit。

TLB 有 4 组 16 项, 所以索引 4 行, TLBI 2bit

TLBT 有  $\text{VPN} - \text{TLBI} = 8 - 2 = 6\text{bit}$

Cache Line  $4\text{B} = 2^2 \text{Byte}$ , 所以 CO 有 2bit.

Cache 有 16 行, 所以 CI = 4bit

$\text{CT} = \text{PA} - \text{CI} - \text{CO} = 12 - 4 - 2 = 6 \text{ bit}$

综上所述:

TLBI: 2 bit, 来自 VA 的[7:6]位

TLBT: 6 bit, 来自 VA 的[13:8]位

VPN: 8 bit, 来自 VA 的[13:6]位

PPN: 6 bit, 来自 PA 的[11:6]位

VPO: 6 bit, 来自 VA 的[5:0]位

PPO: 6 bit, 来自 PA 的[5:0]位

CT: 6 bit, 来自 PA 的[11:6]位

CI: 4 bit, 来自 PA 的[5:2]位

CO: 2 bit, 来自 PA 的[1:0]位

(2).

虚拟地址	0x027c	0x03a9	0x0040	0x03d7
TLB 命中	F	F	F	F
缺页	F	F	T	F
物理页号	0x47c	0x469	--	0x357
缓存命中	F	F	--	T
字节偏移	0	1	0	3
返回的缓存字节	--	--	--	0x1d

简单解析:

0x027c = 0b 000010 01 111100

TLBI = 1, TLBT=2, 查表 invalid, 转到查页表。VPN = 1001 = 9 得到 PPN = 17  
所以物理地址 = 0b 010001 1111 00. CI=15, CT=17, 查表发现 Tag 不同, 所以  
Cache miss.

0x03a9 = 0b 000011 10 101001

TLBI = 2, TLBT=3, 查表 invalid, 转到查页表。VPN = 1110 = 14 得到 PPN = 11  
所以物理地址 = 0b 010001 1010 01. CI=2, CT=17, CO=1, 查表发现 Tag 不同,  
所以 Cache miss.

0x0040 = 0b 000000 01 000000

TLBI = 1, TLBT=0, 查表 invalid, 转到查页表。VPN = 0001 = 1, 页表无效, 缺  
页异常。

0x03d7 = 0b000011 11 010111

TLBI = 3, TLBT=3, 查表 invalid, 转到查页表。VPN = 1111 = 0F 得到 PPN = 0D  
所以物理地址 = 0b 001101 01 0111. CI=5, CT=0d, CO=3, 查表发现 Tag 相同,  
所以 Cache hit. 得到数据 0b 0001 1101 = 0x0D

16.

使用中段作为索引能带来更高的命中率。

因为程序访存有时空局部性, 所以程序在一段时间内访问的地址可能挨得很近。  
这些地址基本上高段完全相同, 如果采用高段作为索引, 那么在直接映射或 N 路  
组相联的方式下, 每隔一个或 N 个地址就会产生一次 cache miss. 而将中段作为  
index 将减轻这种现象, 带来更高的缓存命中率。

17.

(1).

页面大小  $2048\text{Byte} = 2^{11}\text{Byte}$ , 所以 VPO 位宽 11bit, PPO 位宽也是 11bit.

因为 VA 位宽 32bit, PA 位宽 28bit, 所以 VPN 位宽 21bit, PPN 位宽 17bit.

Cache 块大小  $64\text{Byte} = 2^6\text{Byte}$ , 所以 Cache 块内地址 CO 位宽 6bit

Cache 2 路组相联, 大小 128KB, 有  $128\text{K} / 2 / 64 = 1\text{K} = 2^{10}$  行, 所以 Cache 块号字段位宽 10bit. 因为 Cache 使用物理地址, 所以 Cache 标记字段位宽  $28 - 10 - 6 = 12\text{bit}$

综上所述:

虚页号 21 位

实页号 17 位

Cache 标记字段 12 位

Cache 块号字段 10 位

Cache 块内字段 6 位

(2).

计算后可知 CI 位宽 2bit, CO 位宽 4bit, CT 位宽 22bit

**Program A:**

访问的地址序列为:

0x420, 0x4a0, 0x424, 0x4a4, 0x428, 0x4a8, 0x42c, 0x4ac

上述所有地址的 CI = 0b10. 0x42\_ 的 CT = 0b10000, 0x4a\_ 的 CT = 0b10010. 地址 4Byte 对齐, 所以访问的 8 个数组元素都可以填在缓存行内。所以 Cache 状态为:

	T	V	W0	W1	W2	W3	T	V	W0	W1	W2	W3
0		0						0				
1		0						0				
2	0x10	1	a[0]	a[1]	a[2]	a[3]	0x12	1	a[32]	a[33]	a[34]	a[35]
3		0						0				

Cache 命中率 =  $6 / 8 = 75\%$ .

**Program B:**

访问的地址序列为:

0x420, 0x424, 0x428, 0x42c, 0x4a0, 0x4a4, 0x4a8, 0x4ac

上述所有地址的 CI = 0b10. 0x42\_ 的 CT = 0b10000, 0x4a\_ 的 CT = 0b10010. 地址 4Byte 对齐, 所以访问的 8 个数组元素都可以填在缓存行内。所以 Cache 状态为:

	T	V	W0	W1	W2	W3	T	V	W0	W1	W2	W3
0		0						0				

1		0						0				
2	0x10	1	a[0]	a[1]	a[2]	a[3]	0x12	1	a[32]	a[33]	a[34]	a[35]
3		0						0				

Cache 命中率 =  $6 / 8 = 75\%$ .