1. （20 分）假设一个处理器使用 32 位虚拟地址和 32 位物理地址，处理器缓存参数如下：8 组、2 路组关联、256 字节缓存块大小。处理器系统使用基于 1KB 页面大小的虚拟内存系统。设计者需要选择：是采用虚拟地址寻址方式的 Cache（即虚拟索引和虚拟标记），还是采用物理地址寻址方式的 Cache（即物理索引和物理标记）。

1) 如果虚拟地址 VA = 0x00006825 (0b0000_0000_0000_0000_0110_1000_0010_0101)。在下表中标记出与此虚拟地址对应的数据可能驻留在虚拟寻址和物理寻址高速缓存的位置。如果数据可以驻留在所有条目中，则将"所有条目"填入相应位置。

| 虚拟地址寻址cache | | | | 物理地址寻址cache | | |
|---|---|---|---|---|---|---|
| Index | Way0 | Way1 | | Index | Way0 | Way1 |
| 0 | | | | 0 | | |
| 1 | | | | 1 | | |
| 2 | | | | 2 | | |
| 3 | | | | 3 | | |
| 4 | | | | 4 | | |
| 5 | | | | 5 | | |
| 6 | | | | 6 | | |
| 7 | | | | 7 | | |

**页偏移地址：1K 页面，最低 10 位地址（bit9~bit0：00_0010_0101）；**
**Cache 索引地址：bit10 ~ bit8（8 组），因为 Cache 行是 256 字节，行内便宜用 bit7 ~ bit0 来寻址；**
**因此，索引地址的最高位可能会根据 VA→PA 转换过程发生变化。**

| 虚拟地址寻址cache | | | | 物理地址寻址cache | | |
|---|---|---|---|---|---|---|
| Index | Way0 | Way1 | | Index | Way0 | Way1 |
| 0 | X | X | | 0 | X | X |
| 1 | | | | 1 | | |
| 2 | | | | 2 | | |
| 3 | | | | 3 | | |
| 4 | | | | 4 | X | X |
| 5 | | | | 5 | | |
| 6 | | | | 6 | | |
| 7 | | | | 7 | | |

2) 如果物理地址：PA = 0x00006825 (0b0000_0000_0000_0000_0110_1000_0010_0101)。在下表中标记出与此物理地址对应的数据可能驻留的位置。如果数据可以驻留在所有条目中，则将“所有条目”填入相应位置。

| 虚拟地址寻址cache | | | | 物理地址寻址cache | | |
|---|---|---|---|---|---|---|
| Index | Way0 | Way1 | | Index | Way0 | Way1 |
| 0 | | | | 0 | | |
| 1 | | | | 1 | | |
| 2 | | | | 2 | | |
| 3 | | | | 3 | | |
| 4 | | | | 4 | | |
| 5 | | | | 5 | | |
| 6 | | | | 6 | | |
| 7 | | | | 7 | | |

这个问题是把上一问反过来说，对应同一个物理地址，其虚拟地址的 bit10 可能会改变

| 虚拟地址寻址cache | | | | 物理地址寻址cache | | |
|---|---|---|---|---|---|---|
| Index | Way0 | Way1 | | Index | Way0 | Way1 |
| 0 | X | X | | 0 | X | X |
| 1 | | | | 1 | | |
| 2 | | | | 2 | | |
| 3 | | | | 3 | | |
| 4 | X | X | | 4 | | |
| 5 | | | | 5 | | |
| 6 | | | | 6 | | |
| 7 | | | | 7 | | |

3) 比较上述两种寻址 Cache 方式的差异，是否可以确定采用哪种寻址方式更好？简述理由。

**同一个地址的数据可能映射到多个 Cache 块内，则可能会降低 Cache 缺失的概率，但带来的代价是需要更复杂的替换策略；对于固定容量的 Cache，当映射方式更灵活时，则 Cache 内部存放的空间局部性数据就少，也可能会增加 Cache 缺失的概率。因此不能完全确定那种方案更好。**

2 （20 分）一个 ISA 支持 8 位的、字节可寻址（byte-addressable）的虚拟地址空间。相应的物理内存仅有 128 字节。每个页 16 字节。使用了一个简单的单级转换方案，并且页表存储在物理内存中，物理内存的初始内容如下图所示。

| Frame Number | Frame Contents |
|---|---|
| 0 | Empty |
| 1 | Page 13 |
| 2 | Page 5 |
| 3 | Page 2 |
| 4 | Empty |
| 5 | Page 0 |
| 6 | Empty |
| 7 | Page Table |

使用 LRU 替换策略的 TLB（3-entry）添加到该系统中。最初，这个 TLB 包含了页 0、2 和 13。对于以下仿存（memory references）序列，请在 TLB 命中的 entry 周围画一个圆圈，并在产生错页（page fault）的条目周围画一个矩形。这个访存序列的 TLB 命中率是多少？（注意：在物理内存中使用替换页面的 LRU 策略。）

References (to pages): 0, 13, 5, 2, 14, 14, 13, 6, 6, 13, 15, 14, 15, 13, 4, 3.

References (to pages): (0), (13), 5, 2, [14], (14), 13, [6], (6), (13), [15], 14, (15), (13), [4], [3].

TLB Hit Rate = 7/16

(a) 在这个序列结束时，TLB 中包含了哪三个条目?

4, 13, 3

(b) 8 个物理页（physical pages）的内容是什么?

Pages 14, 13, 3, 2, 6, 4, 15, Page table

3 （20 分）一个具有完美 LRU 替换策略的 2 路组相联（2-way set associative）写回（writeback）缓存需要 $15 \times 2^9$ 位的存储空间来实现标签存储（包括有效位、脏位和 LRU 位）。缓存是虚拟索引、物理标记（virtually indexed physically tagged）。虚拟地址空间为 1 MB，页面大小为 2 KB，缓存块大小为 8 字节。

(a) 缓存的数据存储（data array）大小是多少字节?

8 KB

The cache is 2-way set associative.

So, each set has 2 tags, each of size *t*, 2 valid bits, 2 dirty bits and 1 LRU bit (because a single bit is enough to implement perfect LRU for a 2-way set associative cache).

Let i be the number of index bits.

Tag store size = $2^i \times (2 \times t + 2 + 2 + 1) = 15 \times 2^9$

Therefore,

$2t = 10 \Rightarrow t = 5$

$i = 9$

Data store size = $2^i \times (2 \times 8)$ bytes = $2^9 \times (2 \times 8)$ bytes = 8 KB.

(b) 虚拟索引中有多少位来自虚拟页号?

1 bit

Page size is 2 KB. Hence, the page offset is 11 bits (bits 10:0).

The cache block offset is 3 bits (bits 2:0) and the virtual index is 9 bits (bits 11:3).

Therefore, one bit of the virtual index (bit 11) comes from the virtual page number.

(c) 这个内存系统的物理地址空间有多大?

回答以上问题，并给出简单推导过程。

64 KB

The page offset is 11 bits.

The physical frame number, which is the same as the physical tag is 5 bits.

Therefore, the physical address space is $2^{(11+5)} = 2^{16}$ bytes = 64 KB.

4 （20 分）假设有一个处理器，其中指令是基于 8 字节的操作数（operands）。指令也是用 8 字节编码。假设设计的处理器实现了一个 16 KB 的 4 路组关联缓存（4-way set associative cache），包含 1024 组（set）。这个缓存的效能如何？请解释理由，并展示计算过程。

1) The cache requires two accesses to be effective.

2) The cache cannot exploit spatial locality.

Explanation: The cache has 4×1024 = 4096 cache lines in total. That means, each cache line is 16KB/4096 = 4 bytes. With 4-byte cache lines, each operand and each instruction needs to be stored in two cache lines, which will require 2 accesses to the cache for each load/store operation and instruction fetches. The cache cannot exploit spatial locality, but only can provide benefit by exploiting temporal locality (albeit requiring two accesses).

5（10 分）（Rowbuffer）对于以下内存访问模式，估计在两种不同的调度机制下每次内存访问完成的时间：open-page 和 close-page。假设可以重新排序已经在内存控制器中等待的请求。访问模式只指定被访问的行（row）。所有访问都是针对同一个 bank。假设总线延迟为零。假设 bank 在时间 0 已经预充电（pre-charge）。假设预充电需要 20 ns，加载行缓冲区（activation）需要 20 ns，缓存行传输到输出引脚也需要 20 ns（换句话说，行缓冲区命中（Rowbuffer-hit）需要 20 ns，空行访问（Rowbuffer-empty）需要 40 ns，行缓冲区冲突（Rowbuffer-conflict）需要 60 ns）。

| 访问行 | 到达内存控制器时间 | Open-page | Close-page |
|---|---|---|---|
| X | 10 ns | 50 ns | 50 ns |
| X | 75 ns | 95 ns | 115 ns |
| Y | 100 ns | 160 ns | 175 ns |
| X | 190 ns | 250 ns | 235 ns |
| X | 280 ns | 300 ns | 320 ns |
| Y | 290 ns | 360 ns | 380 ns |

6. （10 分）（Memory hierarchy）考虑一个系统，它有两个处理器插槽（socket）；每个插槽有 6 个 DDR4 内存通道（memory channel）。每个通道最多可以容纳 4 ranks。假设你今天可以购买容量为 2Gb、4Gb 或 8Gb 的 DRAM 芯片。假设这些芯片可以有 4、8 或 16 的数据输出宽度（data output width）。这个内存系统可以支持的最大容量是多少？如果每个内存通道以 1.2 GHz 的频率运行，系统支持的内存带宽是多少？

2 sockets x 6 channels x 4 ranks x 16 chips x 8Gb capacity = 768 GB

2 sockets x 6 channels x 1.2G (cycles per second) x 2 (DDR, hence 2 transfers per cycle) x 64 (bits per transfer) = 230.4GB/s