1. Virtual Memory and Caches [50 points]

Assume that we have a byte-addressable processor that implements paging-based virtual memory using a **three-level** hierarchical page table organization. The virtual address is **46 bits**, and the physical memory is **4GB**. The page table base register (PTBR) stores the base address of the first-level table (*PT1*). All the page tables have the same size of **one physical page** for the first-level (*PT1*), second-level (*PT2*), and third level (*PT3*). Each *PT1*/*PT2* entry stores the base address of a second/third-level table (*PT2*/*PT3*). In contrast to *PT1* and *PT2*, each *PT3* entry stores a page table entry (PTE). The PTE is 4-bytes in size.

The processor has a 64KB **virtually-indexed physically-tagged (VIPT)** L1 cache that is direct mapped with a cache line size of 128 bytes, and a 64-entry TLB.

(a) What is the physical page size? Show all your work.

1. First, the physical address space is $\log_2 4GB = 32bits$. So each PT1 and PT2 entry stores 32bits (4bytes).
2. Since the virtual address space is 46 bits and assume the page size is P, then there are $\frac{2^{46}}{P}$ possible mappings that the page tables can store.
3. Each page table can store $\frac{P}{4}$ mappings because each entry is 4 bytes and the table is P bytes. The three-level hierarchical page table can in total store $(\frac{P}{4})(\frac{P}{4})(\frac{P}{4}) = \frac{2^{46}}{P}$ mappings.
4. $P = 2^{13}$.

(b) How many bits of the virtual page number are used to index the L1 cache?

The cache requires $\log_2 64KB = 16$ bits from the address for indexing. So 3 bits are overlapped with the VPN.

(c) What kind of aliasing problem does this processor's L1 cache have?

Synonym – multiple different virtual addresses map to the same physical address, causing multiple copies of the data belonging to the same physicall address residing in the cache.

(d) In lecture, we learned multiple techniques to resolve this particular aliasing problem (in part (c) above) in a VIPT cache. One of them is to increase the associativity of the cache. To address this aliasing problem for this processor's VIPT cache, what is the minimum associativity that is required for the cache? Show your work.

$2^3 = 8$

(e) We also learned another technique that searches all possible sets that can contain the same physical block. For this VIPT cache, how many sets need to be searched to fix the aliasing problem? Show your work.

$2^3 = 8$