Homework1 memory

Assuming a 64-bit machine with 64-bit OS and the page tables have four levels. level-4 (L4) is PGD, L3 is PUD, L2 is PMD, L1 is PTE.

1. On a 32-bit machine, the huge page size is 4M. On a 64-bit machine, the size of a huge page is 2M. Why not the same size?
2. Why OS/MMU use multi-level page tables to map the virtual address to physical address? What is the maximum spatial overhead for the 4-level page table?
3. In ARM-smmu architecture, how can mmu distinguish whether it is a page entry or invalid entry
4. Please give the advantages and the disadvantages of using block entry / huge page, and give one scenario for each case.
5. Memory attribution bit AP and UXN in page entry can already isolate the kernel space and user space, so why ARM-smmu architecture still needs two ttbr registers (Translation Table Base Register), please give a scenario that two ttbr registers can protect the os but attribution-based isolation can not.
6. TLB (Translation Lookaside Buffer) can cache the translation from a virtual address to a physical address. However, TLB will be flushed after a context switch between processes. Why? Is it necessary to flush TLB when switching between a user application and the OS kernel? Why?
7. Before ARMv8 architecture, there is no DBM(Dirty Bit Modifier) bit in memory attribution. That’s means hardware does not support dirty page. So how can os simulate this procedure and record the dirty page? Please give a possible solution.