Q1

在 MESI 协议中，不同缓存之间需要保持一致的状态，特别是当某个缓存的行处于 M（Modified）状态时，其它缓存中同一地址的缓存行必须处于 I（Invalid）状态，否则会导致不一致。分析表格中不同缓存的状态，特别是对 Tag 为 0x5FFFFF 的行进行检查，因为它在多个缓存中出现且有不同的状态。

1. Cache 0, Set 0：0x5FFFFF，状态为 M
2. Cache 0, Set 1：0x5FFFFF，状态为 E
3. Cache 0, Set 2：0x5FFFFF，状态为 S
4. Cache 1, Set 2：0x5FFFFF，状态为 S
5. Cache 2, Set 2：0x5FFFFF，状态为 S

根据 MESI 协议，0x5FFFFF 地址对应的行在 Cache 0 的 Set 0 中标记为 M 状态时，其他缓存中该地址对应的缓存行应当是 I（Invalid）状态，但实际上 Cache 0、Cache 1 和 Cache 2 中都将它标记为了 S 或 E 状态。

因此，Cache 0 的 Set 0 的行（Tag 为 0x5FFFFF，状态为 M）因为宇宙射线的影响而导致了不一致。

Q2

在这条指令序列中，通过逐条分析每条指令在 MESI 协议下的影响，可以看出系统的缓存一致性不会被破坏。首先，在初始状态下，0x5FFFFF40 位于 Processor 0 的缓存中并处于 M（Modified）状态，这表明它已被修改，且在其他缓存中对应的缓存行应处于 I（Invalid）状态，从而确保缓存一致性。指令 1 是由 Processor 2 执行的 ld 0x51110040，在此时 0x51110040 位于 Processor 2 缓存中并处于 E（Exclusive）状态，因此读取该地址的数据不会引起任何状态转换，也不会产生不一致性。

接下来是指令 2，由 Processor 0 执行的 st 0x5FFFFF40 操作。这一写操作不会影响 MESI 状态，因为 0x5FFFFF40 已经在 Processor 0 的缓存中处于 M 状态，这意味着它可以直接写入且不影响其他处理器的缓存。指令 3 由 Processor 3 执行的是 st 0x51110040，该操作将 Processor 3 缓存中 0x51110040 从 S（Shared）状态转变为 M（Modified）状态。根据 MESI 协议，这一操作将导致 Processor 2 的 0x51110040 缓存行状态变为 I，因为当一个缓存行进入 M 状态时，其他处理器必须使其缓存中的相同行无效，以维护数据一致性。

在指令 4 中，Processor 1 执行 ld 0x5FFFFF80。该地址在 Processor 1 的缓存中处于 E 状态，读取操作不会引起状态改变，也不会导致不一致性。在指令 5 中，Processor 1 执行 ld 0x51110040，由于 Processor 3 在指令 3 中已将 0x51110040 设置为 M 状态，Processor 1 必须将其缓存中对应的行无效化，并从 Processor 3 或主存获取最新数据，从而避免了读取不正确的数据。最后，在指令 6 中，Processor 1 执行 ld 0x5FFFFF40，因为 0x5FFFFF40 在 Processor 0 的缓存中处于 M 状态，Processor 1 的缓存会无效化其缓存行，并从 Processor 0 获取该数据，从而确保数据正确性。

综上所述，在该指令序列下，MESI 协议有效地保证了缓存一致性，各处理器在执行读写操作时均可以获取到正确的数据，因此此指令顺序不会导致错误的执行结果。

Q3

在本题中，我们通过最少的内存访问指令序列实现了系统从初始状态到目标状态的转换，以满足 MESI 协议的一致性要求。首先，Processor 3 对地址 0x5FF00040 执行 st（写入）操作，使得 Processor 3 缓存 Set 0 中的 0x5FF000 的状态从 E（Exclusive）转换为 M（Modified）。根据 MESI 协议的规则，这一操作会导致其他缓存中相同地址的行变为无效（Invalid），因此 Processor 1 和 Processor 0 中的 0x5FF000 将进入 I 状态，从而符合目标状态要求。

接下来，Processor 0 对地址 0x5FFFFF80 执行 ld（读取）操作，使得 Processor 0 缓存 Set 3 中的 0x5FFFFF 进入 E（Exclusive）状态。这是因为当前没有其他处理器对此地址拥有 M 或 S 状态，因此 Processor 0 可以独占该地址的数据。此操作与目标状态一致。

第三步中，Processor 1 对地址 0x5FF00040 执行 ld 操作。由于 Processor 3 在指令 1 中已将 0x5FF000 设置为 M，因此 Processor 1 的缓存行将被置为无效（Invalid），并从 Processor 3 获取数据，从而确保了缓存一致性。最终，Processor 3 对地址 0x51110040 执行 ld 操作，使得 Processor 3 缓存中 0x511100 的状态从 I 变为 S（Shared），并与 Processor 2 共享该地址的数据。此操作确保了 Processor 3 的 Set 1 中 0x511100 处于目标的 S 状态。

综上，通过这四条指令，我们实现了目标状态的最少内存访问。该指令序列有效地利用了 MESI 协议的状态转换规则，在满足系统一致性的同时，确保每个缓存行达到了目标要求。