## CLAIMS

What is claimed is:

1. A method for securely entering a trusted execution environment in a hyper-threading scenario, comprising:

in response to that a logical processor running on a physical processor core generates a trusted execution environment entry event through an approach provided by a virtual machine monitor, the approach comprising executing springboard code provided by the virtual machine monitor, wherein a memory for the springboard code is mapped by a first extended page table used by the logical processor to a first memory address controlled by a guest machine, the first extended page table is configured such that a memory of the trusted execution environment is unmapped or non-executable, the springboard code includes a page table switch instruction that switches the first extended page table to a second extended page table, the springboard code is mapped by the second extended page table to a second memory address controlled by the virtual machine monitor, the second extended page table is configured such that the memory of the trusted execution environment is executable, and the springboard code at the second memory address includes an instruction for labeling a logical processor with a state of expecting to enter a trusted execution environment and an instruction for controlling a logical processor to enter a trusted execution environment, executing the instruction for labeling the logical processor with the state of expecting to enter a trusted execution environment; and

in response to determining that all logical processors corresponding to the physical processor core are labeled with the state of expecting to enter a trusted execution environment, separately controlling each one of the logical processors to enter a trusted execution environment built on the physical processor core.

2. The method according to claim 1, wherein the springboard code includes a hypercall trusted execution environment entry instruction, the virtual machine monitor being

configured to label a state of a logical processor and control the logical processor to enter a trusted execution environment.

3. The method according to claim 1, comprising:

switching the second extended page table to the first extended page table after controlling a logical processor to exit the trusted execution environment built on the physical processor core.

4. The method according to claim 1, comprising:

in response to that a logical processor running on the physical processor core generates a trusted execution environment exit event due to an interrupt, if another logical processor has entered the trusted execution environment built on the physical processor core, or has switched from the first extended page table to the second extended page table but has not yet entered the trusted execution environment built on the physical processor core, injecting an interrupt into the other logical processor after performing the following operations:

in response to that instructions following the page table switch instruction in the springboard code at the first memory address are all no-operation instructions, switching the second extended page table back to the first extended page table;

in response to that instructions following the page table switch instruction in the springboard code at the first memory address include a non-no-operation instruction, switching the second extended page table back to the first extended page table and rewriting a return address of the other logical processor to be a return address corresponding to the first extended page table; and

separately controlling each logical processor to exit the trusted execution environment built on the physical processor core in response to that the logical processors each generate a trusted execution environment exit event.

5. The method according to claim 1, comprising:

configuring a switch condition for switching from a non-root mode to a root mode, the switch condition including detecting that a guest machine executes a trusted execution environment building instruction; and

in response to the switch condition being met, executing a trusted execution environment building instruction to build the trusted execution environment on the physical processor core.

6. The method according to claim 1, wherein the determining that all the logical processors running on the physical processor core are labeled with the state of expecting to enter a trusted execution environment includes at least one of:

determining that all the logical processors running on the physical processor core are labeled as expecting to enter a same trusted execution environment; or

determining that all the logical processors running on the physical processor core are separately labeled as expecting to enter different trusted execution environments, and the logical processors trust the trusted execution environments expected to be entered by one another.

7. The method according to claim 1, comprising:

verifying the virtual machine monitor in response to that all the logical processors running on the physical processor core separately enter the trusted execution environment built on the physical processor core; and

in response to determining that the virtual machine monitor is successfully verified, processing privacy data in the trusted execution environment built on the physical processor core.

8. The method according to claim 7, wherein the verifying the virtual machine monitor includes:

verifying the virtual machine monitor by using a trusted platform module.

9. The method according to claim 1, comprising:

in response to that a logical processor running on the physical processor core generates a trusted execution environment exit event, labeling the logical processor that generates the trusted execution environment exit event with a state of expecting to exit a trusted execution environment, or clearing the state of expecting to enter a trusted execution environment with which the logical processor that generates the trusted execution environment exit event is labeled; and

separately controlling each logical processor to exit the trusted execution environment built on the physical processor core in response to determining that all the logical processors corresponding to the physical processor core are labeled with the state of expecting to exit a trusted execution environment or the state of expecting to enter a trusted execution environment with which all the logical processors corresponding to the physical processor core are labeled has been cleared.

10. The method according to claim 9, wherein the logical processor running on the physical processor core generating the trusted execution environment exit event includes at least one of:

the logical processor running on the physical processor core initiating a trusted execution environment exit hypercall; or

the logical processor running on the physical processor core experiencing an interrupt or exception.

11. The method according to claim 9, wherein the separately controlling each logical processor to exit the trusted execution environment built on the physical processor core includes:

in response to that at least one logical processor running on the physical processor core is labeled with the state of expecting to exit a trusted execution environment or the state of expecting to enter a trusted execution environment with which at least one logical processor running on the physical processor core is labeled has been cleared, and there is a remaining logical processor labeled with the state of expecting to enter a trusted execution environment,

sending an empty interrupt pre-registered by a guest operating system to the remaining logical processor; and

separately controlling each logical processor to exit the trusted execution environment built on the physical processor core after the remaining logical processor is labeled with the state of expecting to exit a trusted execution environment or the state of expecting to enter a trusted execution environment with which the remaining logical processor is labeled has been cleared.

12. An electronic device, comprising:

a processor; and

a memory having executable instructions stored thereon, the executable instructions, when executed by the processor, enable the processor to implement acts including:

in response to that a logical processor running on a physical processor core generates a trusted execution environment entry event through an approach provided by a virtual machine monitor, the approach comprising executing springboard code provided by a virtual machine monitor, a memory for the springboard code is mapped by a first extended page table used by the logical processor to a first memory address controlled by a guest machine, the first extended page table is configured such that a memory of the trusted execution environment is unmapped or non-executable, the springboard code includes a page table switch instruction that switches the first extended page table to a second extended page table, the springboard code is mapped by the second extended page table to a second memory address controlled by the virtual machine monitor, the second extended page table is configured such that the memory of the trusted execution environment is executable, and the springboard code at the second memory address includes an instruction for labeling a logical processor with the state of expecting to enter a trusted execution environment and an instruction for controlling a logical processor to enter a trusted execution environment, executing the instruction for labeling the logical processor with a state of expecting to enter a trusted execution environment; and

in response to determining that all logical processors corresponding to the physical processor core are labeled with the state of expecting to enter a trusted execution environment,

separately controlling each one of the logical processors to enter a trusted execution environment built on the physical processor core.

13. The device according to claim 12, wherein the springboard code includes a hypercall trusted execution environment entry instruction, the virtual machine monitor being configured to label a state of a logical processor and control the logical processor to enter a trusted execution environment.

14. A non-transitory computer-readable storage medium having executable instructions stored thereon, which when executed by a processor, configure the processor to implement acts comprising:

in response to that a logical processor running on a physical processor core generates a trusted execution environment entry event through an approach provided by a virtual machine monitor, the approach comprising executing springboard code provided by a virtual machine monitor, wherein a memory for the springboard code is mapped by a first extended page table used by the logical processor to a first memory address controlled by a guest machine, the first extended page table is configured such that a memory of the trusted execution environment is unmapped or non-executable, the springboard code includes a page table switch instruction that switches the first extended page table to a second extended page table, the springboard code is mapped by the second extended page table to a second memory address controlled by the virtual machine monitor, the second extended page table is configured such that the memory of the trusted execution environment is executable, and the springboard code at the second memory address includes an instruction for labeling a logical processor with a state of expecting to enter a trusted execution environment and an instruction for controlling a logical processor to enter a trusted execution environment, executing the instruction for labeling the logical processor with the state of expecting to enter a trusted execution environment; and

in response to determining that all logical processors corresponding to the physical processor core are labeled with the state of expecting to enter a trusted execution environment,

separately controlling each one of the logical processors to enter a trusted execution environment built on the physical processor core.

15. The storage medium according to claim 14, wherein the acts comprise:

configuring a switch condition for switching from a non-root mode to a root mode, the switch condition including detecting that a guest machine executes a trusted execution environment building instruction; and

in response to the switch condition being met, executing a trusted execution environment building instruction to build the trusted execution environment on the physical processor core.

16. The storage medium according to claim 14, wherein the acts comprise:

in response to that a logical processor running on the physical processor core generates a trusted execution environment exit event, labeling the logical processor that generates the trusted execution environment exit event with a state of expecting to exit a trusted execution environment, or clearing the state of expecting to enter a trusted execution environment with which the logical processor that generates the trusted execution environment exit event is labeled; and

separately controlling each logical processor to exit the trusted execution environment built on the physical processor core in response to determining that all the logical processors corresponding to the physical processor core are labeled with the state of expecting to exit a trusted execution environment or the state of expecting to enter a trusted execution environment with which all the logical processors corresponding to the physical processor core are labeled has been cleared.