configuration of the CPU, the NC group or NC is directly or indirectly pointed to. It does not necessarily mean that cross-domain access exists at a current time point. However, when the CPU sends a cross-domain access request, the access request may pass through the NC group or NC.

Directly pointing to an NC is, for example, directly connecting to the NC that is pointed to. In the CPU route configuration, a next node is the NC that is pointed to. When the CPU sends a cross-domain access request, according to the CPU route configuration, a next node that the cross-domain access request passes through is the NC that is pointed to. Indirectly pointing to an NC is, for example, passing through another node (such as an NC or a CPU) first and then connecting to the NC that is pointed to. When the CPU sends a cross-domain access request, according to the CPU route configuration, the cross-domain access request may pass through other nodes first and then arrive at the NC that is pointed to. The foregoing describes principles of directly or indirectly pointing to an NC, and principles of directly or indirectly pointing to an NC group are the same.

Embodiment 1

Referring to FIG. 3, this embodiment of the present invention provides an NC hot-remove method. This method is described based on the architecture in FIG. 2.

When an NC in the first NC group is to be removed, a main CPU controls all CPUs in the multi-CPU system that perform cross-domain access by using the first NC group, to switch to perform cross-domain access by using a second NC group.

Assuming that the main CPU is a CPU 234 in a QPI domain 23, and an NC to be removed is an NC 211. For ease of description, the to-be-removed NC 211 in this embodiment is called a target NC, the NC group 21 is called a first NC group, and the NC group 22 is called a second NC group.

S301. A main CPU receives a signal indicating that an NC in a first NC group needs to be removed.

Exemplarily, when a user is ready to remove the NC 211, the user may send a remove instruction to the main CPU 231 by using an operating system, where the remove instruction includes an identifier of the NC 211. In this way, when receiving the remove instruction, the main CPU 231 may determine that an NC to be removed exists in the NC group 21.

Exemplarily, after detecting that reliability of the NC 211 in the NC group 21 is reduced, the CPU 234 may start removing the NC 211. After a route that passes through the NC group 21 is switched to an NC group 22 and use of the NC group 21 is stopped, the NC 211 may be hot-removed.

S302. The main CPU instructs a CPU that performs cross-domain access by using the first NC group, to configure a data state recorded by the CPU to an invalid state.

The main CPU may learn a topological relationship between CPUs in a multi-CPU system, or in

CLAIMS

1. A CPU, serving as a main CPU (234) that manages a multi-CPU system, characterized in that the multi-CPU system comprises: a first Quick-Path Interconnect, QPI domain (23), a second QPI domain (24), a first node controller NC group (21), and a second NC group (22), the first NC group (21) is connected to the first QPI domain (23) and the second QPI domain (24), the second NC group (22) is connected to the first QPI domain (23) and the second QPI domain, each QPI domain comprises at least one CPU, the main CPU (234) belongs to the first QPI domain or the second QPI domain (24), and the main CPU (234) is configured to:

when an NC in the first NC group (21) is to be removed, control all CPUs in the multi-CPU system that perform cross-domain access by using the first NC group (21), to switch, by modifying a route configuration, to perform cross-domain access by using the second NC group;

after the NC to be removed is removed and a target NC (211) is added to the first NC group (21), establish a topological relationship with an original NC in the first NC group (21); and

control at least one CPU that performs cross-domain access by using the second NC group (22), to switch, depending on different cross-domain sink CPUs that are accessed between domains and by modifying a route configuration, cross-domain access to some cross-domain sink CPUs to cross-domain access performed by using the first NC group (21).

2. The CPU according to claim 1, characterized in that the main CPU is specifically configured to:

instruct the CPUs that perform cross-domain access by using the first NC group (21), to configure a recorded data state to an invalid state;

initiate system silence to configure all CPUs in the multi-CPU system to a silent state;

instruct the CPUs that perform cross-domain access by using the first NC group (21), to configure a port that is in a route register and that accesses a CPU in another domain, to a port that performs cross-domain access by using the second NC group (22); and

initiate cancellation of system silence.

3. The CPU according to claim 2, characterized in that the main CPU (234) being configured to instruct the CPUs that perform cross-domain access by using the first NC group (21), to configure a recorded data state to an invalid state specifically comprises: the main CPU (234) being configured to:

instruct the CPUs that perform cross-domain access by using the first NC group (21), to clear data in a cache, so as to configure a modified state in data states to the invalid state, wherein the data states are recorded by the CPUs that perform cross-domain access by using the first NC group

(21); and

instruct all NCs in the first NC group (21) to broadcast share-state and exclusive-state data states, so as to configure a shared state and an exclusive state in the data states to the invalid state, wherein the data states are recorded by the CPUs that perform cross-domain access by using the first NC group (21).

4. The CPU according to claim 2, characterized in that the main CPU (234) being configured to initiate system silence to configure all CPUs in the multi-CPU system to a silent state specifically comprises: the main CPU (234) being configured to:

make remaining cores except one or more reserved cores in the main CPU (234) silent, and make all cores of other CPUs in the multi-CPU system silent.

5. The CPU according to claim 1, characterized in that the main CPU (234) is further configured to:

after the NC to be removed is removed and a target NC (211) is added to the first NC group (21) and establishes a topological relationship with an original NC in the first NC group (21), control at least one CPU that is in the multi-CPU system and that performs cross-domain access by using the second NC group (21), to switch, by modifying a route configuration, to perform cross-domain access by using the first NC group (21).

6. The CPU according to claim 1 or 2, characterized in that the main CPU (234) is further configured to:

after the NC to be removed is removed and a target NC (211) is added to the first NC group (21), instruct at least one CPU that performs cross-domain access by using the second NC group (22), to configure a recorded data state to the invalid state;

initiate system silence to configure all CPUs in the multi-CPU system to the silent state;

establish a topological relationship between the target NC (211) and an original NC in the first NC group (21);

instruct the at least one CPU to configure a port that is in a route register and that accesses a CPU in another domain, to a port that performs cross-domain access by using the first NC group; and

initiate cancellation of system silence.

7. The CPU according to claim 1, characterized in that the main CPU (234) is further configured to:

after a target NC (211) is added into the first NC group (21), send an instruction to at least one CPU that performs cross-domain access by using the second NC group (22), and instruct the at least

one CPU to set data states corresponding to M cross-domain sink CPUs in recorded data states to an invalid state, wherein the data states recorded by the CPU correspond to a total of N cross-domain sink CPUs, and M<N;

initiate system silence to configure all CPUs in the multi-CPU system to a silent state;

modify a route register configuration of the at least one CPU, so as to configure a port that is in a route register and that accesses the M CPUs, to a port that performs cross-domain access by using the first NC group (21); and

cancel system silence after finishing modifying the route register configuration.

8. A multi-CPU system which is managed by the main CPU according to claim 7, characterized in that

the M CPUs are directly connected to the first NC group (21).

9. The multi-CPU system which is managed by the main CPU according to claim 7, characterized in that the instructing, by the main CPU (234), the at least one CPU to set data states corresponding to M cross-domain sink CPUs in recorded data states to an invalid state specifically comprises: the main CPU (234) being configured to:

instruct the at least one CPU to clear data that is in a cache and from the M CPUs, so as to configure a modified state in the data states corresponding to the M cross-domain sink CPUs to the invalid state; and

instruct an NC in the second NC group (22) to broadcast share-state and exclusive-state data states in the recorded data states corresponding to the M cross-domain sink CPUs, so as to configure a shared state and an exclusive state in the data states corresponding to the M cross-domain sink CPUs to the invalid state.

10. A computer, characterized in that the computer comprises the first QPI domain (23), the second QPI domain (24), the first NC group (21), and the second NC group (22) and the computer is managed by a main CPU according to any one of claims 1 to 9, and the computer further comprises a BIOS, wherein the BIOS is used to store a program required by a main CPU (234) for generating an instruction.

11. A multi-CPU system management method, applicable to a main CPU that manages a multi-CPU system, characterized in that the multi-CPU system comprises: a first Quick-Path Interconnect, QPI domain, a second QPI domain, a first node controller NC group, and a second NC group, the first NC group is connected to the first QPI domain and the second QPI domain, the second NC group is connected to the first QPI domain and the second QPI domain, each QPI domain comprises at least one CPU, the main CPU belongs to the first QPI domain or the second QPI domain, and the method comprises:

when an NC in the first NC group is to be removed, controlling all CPUs in the multi-CPU system that perform cross-domain access by using the first NC group, to switch, by modifying a route configuration, to perform cross-domain access by using the second NC group;

after the NC to be removed is removed and a target NC (211) is added to the first NC group (21), establishing a topological relationship with an original NC in the first NC group (21); and

controlling at least one CPU that performs cross-domain access by using the second NC group (22), to switch, depending on different cross-domain sink CPUs that are accessed between domains and by modifying a route configuration, cross-domain access to some cross-domain sink CPUs to cross-domain access performed by using the first NC group (21).

12. The multi-CPU system management method according to claim 11, characterized in that the controlling all CPUs in the multi-CPU system that perform cross-domain access by using the first NC group, to switch, by modifying a route configuration, to perform cross-domain access by using the second NC group specifically comprises:

instructing the CPUs that perform cross-domain access by using the first NC group, to configure a recorded data state to an invalid state (S302);

initiating system silence to configure all CPUs in the multi-CPU system to a silent state (S303);

instructing the CPUs that perform cross-domain access by using the first NC group, to configure a port that is in a route register and that accesses a CPU in another domain, to a port that performs cross-domain access by using the second NC group (S304); and

initiating cancellation of system silence (S305).

13. The multi-CPU system management method according to claim 12, characterized in that the instructing the CPUs that perform cross-domain access by using the first NC group, to configure a recorded data state to an invalid state specifically comprises:

instructing the CPUs that perform cross-domain access by using the first NC group, to clear data in a cache, so as to configure a modified state in data states to the invalid state, wherein the data states are recorded by the CPUs that perform cross-domain access by using the first NC group; and

instructing all NCs in the first NC group to broadcast share-state and exclusive-state data states, so as to configure a shared state and an exclusive state in the data states to the invalid state, wherein the data states are recorded by the CPUs that perform cross-domain access by using the first NC group.

14. The multi-CPU system management method according to claim 12, characterized in that the initiating system silence to configure all CPUs in the multi-CPU system to a silent state

specifically comprises:

making remaining cores except one or more reserved cores in the main CPU (234) silent, and making all cores of other CPUs in the multi-CPU system silent.

15. The multi-CPU system management method according to claim 11, characterized in that the method further comprises:

after the NC to be removed is removed and a target NC (211) is added to the first NC group (21) and establishes a topological relationship with an original NC in the first NC group (21), controlling at least one CPU that is in the multi-CPU system and that performs cross-domain access by using the second NC group (22), to switch to perform cross-domain access by using the first NC group (21).

16. The multi-CPU system management method according to claim 11 or 12, characterized in that the method further comprises:

after a target NC (211) is added to the first NC group (21), instructing at least one CPU that performs cross-domain access by using the second NC group (22), to configure a recorded data state to the invalid state;

initiating system silence to configure all CPUs in the multi-CPU system to the silent state;

establishing a topological relationship between the target NC (211) and an original NC in the first NC group (21);

instructing the at least one CPU to configure a port that is in a route register and that accesses a CPU in another domain, to a port that performs cross-domain access by using the first NC group (21); and

initiating cancellation of system silence.

17. The multi-CPU system management method according to claim 11, characterized in that the method further comprises:

after a target NC (211) is added into the first NC group (21), sending an instruction to at least one CPU that performs cross-domain access by using the second NC group (22), and instructing the at least one CPU to set data states corresponding to M cross-domain sink CPUs in recorded data states to an invalid state, wherein the data states recorded by the CPU correspond to a total of N cross-domain sink CPUs, and $M<N$;

initiating system silence to configure all CPUs in the multi-CPU system to a silent state;

modifying a route register configuration of the at least one CPU, so as to configure a port that is in a route register and that accesses the M CPUs, to a port that performs cross-domain access by using the first NC group (21); and

canceling system silence after finishing modifying the route register configuration.

18. A CPU, serving as a main CPU (234) that manages a multi-CPU system, characterized in that the multi-CPU system comprises: a first Quick-Path Interconnect, QPI domain, a second QPI domain, a first node controller NC group (21), and a second NC group (22), the first NC group (21) is connected to the first QPI domain (23) and the second QPI domain (24), the second NC group (22) is connected to the first QPI domain (23) and the second QPI domain (24), each QPI domain comprises at least one CPU, the main CPU (234) belongs to the first QPI domain (23) or the second QPI domain (24), and the main CPU (234) is configured to:

when a target NC (211) in the first NC group (21) is to be removed, control all CPUs in the multi-CPU system that perform cross-domain access by using the target NC (211), to switch, by modifying a route configuration, to perform cross-domain access by using the second NC group (22);

after the NC to be removed is removed and a target NC (211) is added to the first NC group (21), establish a topological relationship with an original NC in the first NC group (21); and

control at least one CPU that performs cross-domain access by using the second NC group (22), to switch, depending on different cross-domain sink CPUs that are accessed between domains and by modifying a route configuration, cross-domain access to some cross-domain sink CPUs to cross-domain access performed by using the first NC group (21).

19. The CPU according to claim 18, characterized in that the main CPU (234) being configured to control all CPUs in the multi-CPU system that perform cross-domain access by using the target NC (211), to switch, by modifying a route configuration, to perform cross-domain access by using the second NC group (22) specifically comprises: the main CPU (234) being configured to:

instruct the CPUs that perform cross-domain access by using the target NC (211), to configure a recorded data state to an invalid state;

initiate system silence to configure all CPUs in the multi-CPU system to a silent state;

instruct remaining NCs in the first NC group (21) to update a route configuration to isolate the target NC (211), and instruct the CPUs that perform cross-domain access by using the target NC (211), to configure a port that is in a route register and that accesses a CPU in another domain, to a port that performs cross-domain access by using the second NC group (22); and

initiate cancellation of system silence.

20. The CPU according to claim 19, characterized in that the main CPU (234) being configured to instruct the CPUs that perform cross-domain access by using the target NC (211), to configure a recorded data state to an invalid state specifically comprises: the main CPU (234) being configured to:

instruct the CPUs that perform cross-domain access by using the target NC (211), to clear a cache, so as to configure a modified state in data states to the invalid state, wherein the data states are recorded by the CPUs that perform cross-domain access by using the target NC (211); and

instruct the target NC (211) to broadcast share-state and exclusive-state data states, so as to configure a shared state and an exclusive state in the data states to the invalid state, wherein the data states are recorded by the CPUs that perform cross-domain access by using the target NC (211).

21. A computer, characterized in that the computer comprises the first Quick-Path Interconnect, QPI domain (23), the second QPI domain (24), the first NC group (21), and the second NC group (21), and the computer is managed by a main CPU according to any one of claims 18 to 20, and the computer further comprises a BIOS, wherein the BIOS is used to store a program required by a main CPU (234) for generating an instruction.

22. A multi-CPU system management method, which is used by a main CPU to manage a multi-CPU system, characterized in that the multi-CPU system comprises: a first Quick-Path Interconnect, QPI domain, a second QPI domain, a first node controller NC group, and a second NC group, the first NC group is connected to the first QPI domain and the second QPI domain , the second NC group is connected to the first QPI domain and the second QPI domain, each QPI domain comprises at least one CPU, the main CPU belongs to the first QPI domain or the second QPI domain, and the method comprises:

when a target NC in the first NC group is to be removed, controlling all CPUs in the multi-CPU system that perform cross-domain access by using the target NC, to switch, by modifying a route configuration, to perform cross-domain access by using the second NC group;

after the NC to be removed is removed and a target NC (211) is added to the first NC group (21), establishing a topological relationship with an original NC in the first NC group (21); and

controlling at least one CPU that performs cross-domain access by using the second NC group (22), to switch, depending on different cross-domain sink CPUs that are accessed between domains and by modifying a route configuration, cross-domain access to some cross-domain sink CPUs to cross-domain access performed by using the first NC group (21).

23. The multi-CPU system management method according to claim 22, characterized in that the controlling all CPUs in the multi-CPU system that perform cross-domain access by using the target NC, to switch, by modifying a route configuration, to perform cross-domain access by using the second NC group specifically comprises:

instructing the CPUs that perform cross-domain access by using the target NC, to configure a recorded data state to an invalid state (502);

initiating system silence to configure all CPUs in the multi-CPU system to a silent state (503);

instructing remaining NCs in the first NC group to update a route configuration to isolate the target NC (504), and instructing the CPUs that perform cross-domain access by using the target NC, to configure a port that is in a route register and that accesses a CPU in another domain, to a port that performs cross-domain access by using the second NC group (505); and

initiating cancellation of system silence (506).

24. The multi-CPU system management method according to claim 22, characterized in that the instructing, by the main CPU, the CPUs that perform cross-domain access by using the target NC, to configure a recorded data state to an invalid state (502) specifically comprises:

instructing the CPUs that perform cross-domain access by using the target NC, to clear a cache, so as to configure a modified state in data states to the invalid state, wherein the data states are recorded by the CPUs that perform cross-domain access by using the target NC; and

instructing the target NC to broadcast share-state and exclusive-state data states, so as to configure a shared state and an exclusive state in the data states to the invalid state, wherein the data states are recorded by the CPUs that perform cross-domain access by using the target NC.

25. A CPU, serving as a main CPU that manages a multi-CPU system, characterized in that the multi-CPU system comprises: a first Quick-Path Interconnect, QPI domain (23), a second QPI domain (24), a first node controller NC group, and a second NC group, the first NC group is connected to the first QPI domain (23) and the second QPI domain (24), the second NC group is connected to the first QPI domain (23) and the second QPI domain (24), each QPI domain comprises at least one CPU, the main CPU belongs to the first QPI domain (23) or the second QPI domain (24), and the main CPU is configured to:

when a target NC (211) is added to the first NC group, control at least one CPU that is in the first QPI domain (23) and that performs cross-domain access by using the second NC group, to switch, by modifying a route configuration, to perform cross-domain access by using the target NC (211), wherein the target NC (211) is connected to the first QPI domain (23);

wherein the main CPU further is configured to:

send an instruction to the at least one CPU that performs cross-domain access by using the second NC group, and instruct the at least one CPU to set data states corresponding to M cross-domain sink CPUs in recorded data states to an invalid state, wherein the data states recorded by the CPU correspond to a total of N cross-domain sink CPUs, and $M<N$; and initiate system silence to configure all CPUs in the multi-CPU system to a silent state; and

instruct the at least one CPU to modify a route configuration, so as to configure a port that is in a route register and that accesses the M CPUs, to a port that performs cross-domain access by using the target NC (211).

26. The CPU according to claim 25, characterized in that the main CPU being configured to control at least one CPU that is in the first QPI domain (23) and that performs cross-domain access by using the second NC group, to switch, by modifying a route configuration, to perform cross-domain access by using the target NC (211) specifically comprises: the main CPU being configured to:

instruct the at least one CPU to configure a data state to an invalid state;

initiate system silence to configure all CPUs in the multi-CPU system to a silent state;

instruct the target NC (211) to establish a topological relationship with an original NC in the first NC group;

instruct the at least one CPU to modify a route configuration, so as to modify a port that is in a route register and that accesses a CPU in another domain, to a port that performs cross-domain access by using the target NC (211); and

initiate cancellation of system silence.

27. The CPU according to claim 25, characterized in that the main CPU being configured to instruct the at least one CPU to set data states corresponding to M cross-domain sink CPUs in recorded data states to an invalid state specifically comprises: the main CPU being configured to:

instruct the at least one CPU to clear data that is in a cache and from the M CPUs, so as to configure a modified state in the data states corresponding to the M cross-domain sink CPUs to the invalid state; and

further configured to instruct an NC in the second NC group to broadcast share-state and exclusive-state data states in the recorded data state corresponding to the M cross-domain sink CPUs, so as to configure a shared state and an exclusive state in the data states corresponding to the M cross-domain sink CPUs to the invalid state.

28. A computer, characterized in that the computer comprises the first Quick-Path Interconnect, QPI domain (23), the second QPI domain (24), the first NC group (21), and the second NC group (22) and the computer is managed by a main CPU according to any one of claims 25 to 27, and the computer further comprises a BIOS, wherein the BIOS is used to store a program required by a main CPU (234) for generating an instruction.

29. A multi-CPU system management method, which is used by a main CPU to manage a multi-CPU system, characterized in that the multi-CPU system comprises: a first Quick-Path Interconnect, QPI domain, a second QPI domain, a first node controller NC group, and a second NC group, the first NC group is connected to the first QPI domain and the second QPI domain, the second NC group is connected to the first QPI domain and the second QPI domain, each QPI

domain comprises at least one CPU, the main CPU belongs to the first QPI domain or the second QPI domain, and the method comprises:

when a target NC is added to the first NC group, controlling at least one CPU that is in the first QPI domain and that performs cross-domain access by using the second NC group, to switch, by modifying a route configuration, to perform cross-domain access by using the target NC, wherein the target NC is connected to the first QPI domain;

wherein the controlling at least one CPU that is in the first QPI domain and that performs cross-domain access by using the second NC group, to switch, by modifying a route configuration, to perform cross-domain access by using the target NC specifically comprises::

sending an instruction to the at least one CPU that performs cross-domain access by using the second NC group, and instructing the at least one CPU to set data states corresponding to M cross-domain sink CPUs in recorded data states to an invalid state, wherein the data states recorded by the CPU correspond to a total of N cross-domain sink CPUs, and $M<N$;

initiating system silence to configure all CPUs in the multi-CPU system to a silent state; and

instructing the at least one CPU to modify a route configuration, so as to configure a port that is in a route register and that accesses the M CPUs, to a port that performs cross-domain access by using the target NC.

30. The multi-CPU system management method according to claim 29, characterized in that the controlling at least one CPU that is in the first QPI domain and that performs cross-domain access by using the second NC group, to switch, by modifying a route configuration, to perform cross-domain access by using the target NC specifically comprises:

instructing the at least one CPU to configure a data state to an invalid state (S602);

initiating system silence to configure all CPUs in the multi-CPU system to a silent state (S603);

instructing the target NC to establish a topological relationship with an original NC in the first NC group (S604);

instructing the at least one CPU to modify a route configuration, so as to modify a port that is in a route register and that accesses a CPU in another domain, to a port that performs cross-domain access by using the target NC (S605); and

initiating cancellation of system silence (S606).

.

31. The multi-CPU system management method according to claim 29, characterized in that the instructing the at least one CPU to set data states corresponding to M cross-domain sink CPUs in recorded data states to an invalid state specifically comprises:

instructing the at least one CPU to clear data that is in a cache and from the M CPUs, so as to configure a modified state in the data states corresponding to the M cross-domain sink CPUs to the invalid state; and

further instructing an NC in the second NC group to broadcast share-state and exclusive-state data states in the recorded data state configuration of data of the M CPUs, wherein the recorded data state configuration corresponds to the M cross-domain sink CPUs, so as to configure a shared state and an exclusive state in the data states corresponding to the M cross-domain sink CPUs to the invalid state.