Claims

~~1. A method for executing instructions in a central processing unit (CPU), comprising:~~

~~fetching, from a current thread queue, instructions to form an instruction block and transmitting the instruction block to a CPU execution unit for execution, wherein the instruction block contains one jump instruction, and the jump instruction is the last instruction in the instruction block;~~

~~replenishing the current thread queue with at least one instruction to form a thread queue to be executed;~~

~~determining a target instruction of the jump instruction according to an execution result of the CPU execution unit;~~

~~determining whether the target instruction is contained in the thread queue to be executed; and~~

~~if the target instruction is not contained in the thread queue to be executed, flushing the thread queue to be executed, obtaining the target instruction and adding the target instruction to the thread queue to be executed.~~

~~2. The method according to claim 1, wherein fetching, from a current thread queue, instructions to form an instruction block comprises:~~

~~reading a predetermined threshold number of instructions from the current thread queue, the predetermined threshold number depending on the number of CPU execution units;~~

~~determining whether the jump instruction is present in the predetermined threshold number of instructions; and~~

~~if the jump instruction is present, taking the jump instruction and preceding instruction(s) thereof as the instruction block.~~

~~3. The method according to claim 1, wherein replenishing the current thread queue with at least one instruction to form a thread queue to be executed comprises:~~

~~according to a predicted branch of an instruction prediction, replenishing the current thread queue with at least one instruction corresponding to the predicted branch.~~

~~4. The method according to claim 1, wherein replenishing the current thread queue with at least one instruction to form a thread queue to be executed comprises:~~

~~replenishing the current thread queue with corresponding instruction(s) read from a decoded cache, wherein the decoded cache stores a plurality of pre-fetched and decoded instructions.~~

~~5. The method according to claim 1, wherein the jump instruction is a register operation instruction, and wherein the instruction block further includes at least one memory operation instruction.~~

~~6. The method according to claim 5, wherein determining a target instruction of the jump instruction according to an execution result of the CPU execution unit comprises:~~

~~determining the target instruction of the jump instruction before execution of the at least one memory operation instruction is completed.~~

~~7. The method according to claim 1, wherein obtaining the target instruction comprises:~~

~~determining whether the target instruction is present in a decoded cache, wherein the decoded cache stores a plurality of pre-fetched and decoded instructions;~~

~~if present, obtaining the target instruction from the decoded cache; and~~

~~if not present, obtaining the target instruction from memory.~~

~~8. A central processing unit (CPU) controller, comprising:~~

~~an instruction fetching logic configured to sequentially fetch, from a current thread~~

~~queue, instructions to form an instruction block and transmit the instruction block to a CPU execution unit for execution, wherein the instruction block contains one jump instruction, and the jump instruction is the last instruction in the instruction block;~~

~~an instruction replenishing logic configured to replenish the current thread queue with at least one instruction to form a thread queue to be executed;~~

~~a target determination logic configured to determine a target instruction of the jump instruction according to an execution result of the CPU execution unit;~~

~~a determination logic configured to determine whether the target instruction is contained in the thread queue to be executed; and~~

~~a queue handling logic configured to, if the target instruction is not contained in the thread queue to be executed, flush the thread queue to be executed, obtain the target instruction and add the target instruction to the thread queue to be executed.~~

~~9. A central processing unit (CPU), comprising the controller of claim 8.~~

1. A method for executing instructions in a central processing unit (CPU), comprising:

sequentially fetching, from a current thread queue, a predetermined number of instructions;

determining whether the predetermined number of instructions include a jump instruction;

in response to determining that the predetermined number of instructions does not include the jump instruction, forming a first instruction block containing the predetermined number of instructions;

in response to determining that the predetermined number of instructions includes the jump instruction, forming a second instruction block containing the jump instruction and all instructions preceding the jump instruction in the predetermined number of instructions, such that the second instruction block is formed with the jump instruction as its boundary, and the jump instruction is a last instruction in the second instruction block, wherein the second instruction block containing the jump instruction has fewer instructions than the first instruction block containing no jump instruction;

transmitting the first or second instruction block to a CPU execution unit for execution;

after transmitting the first or second instruction block to the CPU execution unit for execution, replenishing the current thread queue with at least one instruction to form a first thread queue to be executed;

after replenishing the current thread queue with the at least one instruction to form the first thread queue to be executed, suspending renaming and allocation of executable resources for subsequent instructions in the first thread queue to be executed;

after suspending the renaming and the allocation of executable resources, determining a target instruction of the jump instruction according to an execution result of the CPU

execution unit;

determining whether the target instruction of the jump instruction is contained in the first thread queue to be executed;

in response to determining that the target instruction of the jump instruction is not contained in the first thread queue to be executed, flushing the first thread queue to be executed, obtaining the target instruction of the jump instruction and adding the target instruction of the jump instruction to the first thread queue to be executed;

after the target instruction of the jump instruction is added to the first thread queue to be executed, resuming the renaming and the allocation of executable resources for the first thread queue to be executed that includes the target instruction of the jump instruction; and

feeding the first thread queue to be executed to a respective execution unit of the CPU and executing the first thread queue to be executed by the respective execution unit of the CPU.

2. The method according to claim 1, wherein

the predetermined number of instructions is determined based on a number of CPU execution units of the CPU.

3. The method according to claim 1, wherein replenishing the current thread queue with the at least one instruction to form the first thread queue to be executed comprises:

replenishing the current thread queue with one or more instructions from a decoded cache, wherein the decoded cache stores a plurality of pre-fetched and decoded instructions.

4. The method according to claim 1, wherein the jump instruction is a register operation instruction.

5. The method according to claim 4, wherein determining the target instruction of the jump instruction according to the execution result of the CPU execution unit comprises:

determining the target instruction of the jump instruction before execution of the at least one memory operation instruction is completed.

6. The method according to claim 1, wherein obtaining the target instruction of the jump instruction comprises:

determining whether the target instruction of the jump instruction is present in a decoded cache, wherein the decoded cache stores a plurality of pre-fetched and decoded instructions;

in response to determining that the target instruction of the jump instruction is present in the decoded cache, obtaining the target instruction of the jump instruction from the decoded cache; and

in response to determining that the target instruction of the jump instruction is not present in the decoded cache, obtaining the target instruction of the jump instruction from a memory.

7. A central processing unit (CPU), comprising a controller circuit and a plurality of CPU execution units, wherein the controller circuit is communicatively connected to the plurality of CPU execution units and a memory, and the controller circuit comprises:

an instruction fetching logic circuit configured to:

in response to determining that the predetermined number of instructions does not include the jump instruction, form a first instruction block containing the predetermined number of instructions;

in response to determining that the predetermined number of instructions includes the jump instruction, form a second instruction block containing the jump instruction and all instructions preceding the jump instruction in the predetermined number of instructions, such that the second instruction block is formed with the jump instruction as its boundary, and the jump instruction is a last instruction in the second instruction block, wherein the second instruction block containing the jump instruction has fewer instructions than the first instruction block containing no jump instruction; and

transmit the first or second instruction block to a CPU execution unit for execution;

an instruction replenishing logic circuit configured to:

after the first or second instruction block is transmitted to the CPU execution unit for execution, replenish the current thread queue with at least one instruction to form a first thread queue to be executed, and

after the current thread queue is replenished with the at least one instruction to form the first thread queue to be executed, suspend renaming and allocation of executable resources for subsequent instructions in the first thread queue to be executed;

a target determination logic circuit configured to, after suspending the renaming and the allocation of executable resources, determine a target instruction of the jump instruction according to an execution result of the CPU execution unit;

a determination logic circuit configured to determine whether the target instruction of the jump instruction is contained in the first thread queue to be executed; and

a queue handling logic circuit configured to, in response to the target instruction of the jump instruction not being contained in the first thread queue to be executed, flush the first thread queue to be executed, obtain the target instruction of the jump instruction and add the target instruction of the jump instruction to the first thread queue to be executed,

wherein the instruction replenishing logic circuit resumes the renaming and the allocation of executable resources for the first thread queue to be executed that includes the target instruction of the jump instruction;

wherein the instruction fetching logic circuit feeds the first thread queue to be executed to a respective execution unit of the CPU; and

wherein the respective execution unit of the CPU is configured to execute the first thread queue to be executed.

8. The CPU according to claim 7, wherein

the predetermined number of instructions is determined based on a number of the plurality of CPU execution units.

9. The CPU according to claim 7, wherein the instruction replenishing logic circuit is further configured to:

replenish the current thread queue with the at least one instruction corresponding to a predicted branch.

10. The CPU according to claim 7, wherein the instruction replenishing logic circuit is further configured to:

replenish the current thread queue with one or more instructions from a decoded cache, wherein the decoded cache stores a plurality of pre-fetched and decoded instructions.