CLAIMS

1. A task allocation method for use in a chip, the chip comprising a processing core, a microprocessor, and a synchronization signal generation unit, wherein the method comprises:

in response to receiving a synchronization signal, determining, by the microprocessor, whether allocation of a task segment to the processing core is required according to the synchronization signal, wherein the task segment is a part of a task;

in response to the allocation of the task segment to the processing core being required, instructing, by the microprocessor, to allocate the task segment to the processing core;

receiving, by the processing core, the task segment; and

in response to satisfying a first preset condition, sending, by the processing core, a synchronization request signal, wherein generation of the synchronization signal is triggered by the synchronization request signal, the synchronization signal is generated by the synchronization signal generation unit, and the synchronization signal generation unit is realized by hardware or software; wherein the first preset condition comprises: the processing core has received the task segment or executing time taken by the processing core to receive the task segment reaches a preset time threshold;

wherein, in response to receiving the synchronization signal, determining, by the microprocessor, whether the allocation of the task segment to the processing core is required according to the synchronization signal, comprises:

executing, by the microprocessor, its own program;

in response to receiving the synchronization signal, determining, by the microprocessor, the number of synchronization signals received in a synchronization cycle; and

determining, by the microprocessor, the processing core that requires to be allocated the task segment according to the number of the synchronization signals received and a preloaded allocation strategy;

wherein it is preset in the allocation strategy that, in the synchronization cycle, which processing core needs to update the task segment it executes;

wherein the method further comprises:

in response to the allocation of the task segment to the processing core not being required, executing, by the microprocessor, its own program;

executing, by the processing core, a last received task segment; and

in response to satisfying a second preset condition, sending, by the processing core, the synchronization request signal;

wherein the second preset condition is that: the processing core executes a synchronization request signal sending instruction in the last received task segment, or time taken by the processing core to execute the last received task segment reaches a preset time threshold.

2. The task allocation method according claim 1, wherein, instructing, by the microprocessor, to allocate the task segment to the processing core, comprises:

instructing, by the microprocessor, a memory controller to read the task segment from an external memory and send it to the processing core.

3. The task allocation method according to claim 2, wherein, instructing, by the microprocessor, the memory controller to read the task segment from the external memory and send it to the processing core, comprises:

obtaining, by the microprocessor, size of storage space of the processing core; and

instructing, by the microprocessor, the memory controller to read the task segment that matches the size of the storage space of the processing core from the memory and send it to the processing core.

4. The task allocation method according to any one of claims 1-3, wherein the chip comprises at least two processing cores executing the task, and the microprocessor instructs to allocate task segments to the at least two processing cores, respectively.

5. The task allocation method according to claim 4, wherein the synchronization request signal comprises at least two synchronization request signals generated by the at least two processing cores executing the task.

6. The task allocation method according to claim 5, wherein:

generation of the synchronization signal is triggered by the at least two synchronization request signals sent by each of the processing cores executing the task.

7. The task allocation method according to any of claims 1-6, wherein the microprocessor comprises a counter, and after receiving the synchronization signal, the method further comprises:

starting the counter to record the number of the synchronization signals received by the microprocessor.

8. A chip, comprising:

a processing core, configured to execute a task;

a microprocessor, configured to, in response to receiving a synchronization signal, determine whether allocation of a task segment to the processing core is required according to the synchronization signal, wherein the task segment is a part of the task; and configured to, in response to the allocation of the task segment to the processing core being required, instruct to allocate the task segment to the processing core; and

a memory controller, configured to, based on an instruction from the microprocessor, read the task segment corresponding to the instruction from an external memory and send it to the processing core;

wherein the chip is configured to implement the task allocation method according to any one of claims 1-7.

9. An electronic device, comprising: a memory for storing computer-readable instructions; and one or more processors for executing the computer-readable instructions, which, upon execution, cause the processors to implement the task allocation method according to any one of claims 1-7.

10. A non-transitory computer-readable storage medium, the non-transitory computer-readable storage medium storing computer instructions for causing a computer to execute the task allocation method according to any one of claims 1-7.

11. A computer program product, comprising computer instructions, wherein, when the computer instructions are executed by a computing device, the computing device executes the task allocation method according to any one of claims 1-7.

12. A computing apparatus, comprising one or more chips according to claim 8.