

SUMMARY

[0009] According to the Detailed Description, a method for dynamic heterogeneous polymorphic processing in a processor including a plurality of processing cores is presented. Each of the plurality of processing cores is assigned to one of a plurality of core clusters and each of the plurality of core clusters dynamically forms a coalition comprising two or more processing cores of the plurality of processing cores. The method includes the steps of receiving a processing task comprising a plurality of serial threads and dividing each of the plurality of serial threads into a plurality of basic blocks, each of the plurality of basic blocks comprising a single entry point and a single exit point. The method also includes creating a control flow graph for the plurality of basic blocks, determining whether each of the plurality of serial threads requires more than one processing core, and sending a go-into-coalition-mode-now instruction to ones of the plurality of core clusters for handling ones of the plurality of serial threads that require more than one processing core. Further, the method includes the step of adding one of a plurality of sentinel instructions to each of the plurality of basic blocks of the ones of the plurality of serial threads that require more than one processing core, each of the plurality of sentinel instructions comprising information for processing of the plurality of basic blocks. And finally, the method includes sending a go-out-of-coalition-mode-now instruction to ones of the plurality of core clusters upon execution completion of each of the corresponding ones of the plurality of serial threads that requires more than one processing core.

[0010] In accordance with another aspect of the present embodiment, a dynamic heterogeneous polymorphic processor is presented. The processor includes a plurality of processing cores, one or more coalition logic and cluster input control logic.

Each of the plurality of processing cores is assigned to one of a plurality of core clusters and each of the one or more coalition logic is associated with each of the plurality of core clusters for enabling dynamic heterogeneous polymorphic processing by a coalition of two or more of the plurality of processing cores within the associated one of the plurality of core clusters. Each of the one or more coalition logic switches the two or more of the plurality of processing cores into a coalition in response to the one of the plurality of core clusters receiving a coalition request command. The cluster input control logic is associated with each of the plurality of core clusters and receives a processing task comprising a plurality of serial threads, divides each of the plurality of serial threads into a plurality of basic blocks, each of the plurality of basic blocks including a single entry point and a single exit point, and creates a control flow graph for the plurality of basic block. The cluster input control logic associated with each of the plurality of core clusters further determines whether processing of each of the plurality of serial threads requires one processing core or a coalition of processing cores in response to complexity of instruction-level parallelism of each of the plurality of serial threads. Further, the cluster input control logic determines that processing of one of the plurality of serial threads requires two processing cores in response to determining that the one of the plurality of serial threads comprises a medium instruction level parallelism serial thread. The cluster input control logic then sends a go-into-coalition-mode-now instruction to the associated one of the plurality of core clusters for the associated one of the plurality of core clusters to form a coalition of two of the plurality of processing cores of the associated one of the plurality of core clusters and passes the one of the plurality of serial threads to coalition logic associated with the coalition.

[0011] In accordance with a further aspect of the present embodiment, a dynamic heterogeneous polymorphic processor is presented. The processor includes a plurality of processing cores, one or more coalition logic and cluster input control logic. Each of the plurality of processing cores is assigned to one of a plurality of core clusters and each of the one or more coalition logic is associated with each of the plurality of core clusters for enabling dynamic heterogeneous polymorphic processing by a coalition of two or more of the plurality of processing cores within the associated one of the plurality of core clusters. Each of the one or more coalition logic switches the two or more of the plurality of processing cores into a coalition in response to the one of the plurality of core clusters receiving a coalition request command. The cluster input control logic is associated with each of the plurality of core clusters and receives a processing task comprising a plurality of serial threads, divides each of the plurality of serial threads into a plurality of basic blocks, each of the plurality of basic blocks including a single entry point and a single exit point, and creates a control flow graph for the plurality of basic blocks. The cluster input control logic associated with each of the plurality of core clusters further determines whether processing of each of the plurality of serial threads requires one processing core or a coalition of processing cores in response to complexity of instruction-level parallelism of each of the plurality of serial threads. Finally, the cluster input control logic determines that processing of one of the plurality of serial threads requires four processing cores in response to determining that the one of the plurality of serial threads comprises a high instruction level parallelism serial thread, the cluster input control logic sending a go-into-coalition-mode-now instruction to the associated one of the plurality of core clusters for the associated one of the plurality of core clusters to form a coalition of four of the plurality of processing cores of the associated one of the plurality of core clusters and

passing the one of the plurality of serial threads to coalition logic associated with the coalition.

[0012] In accordance with yet a further aspect of the present embodiment, a dynamic heterogeneous polymorphic processor is presented. The processor includes a plurality of processing cores, one or more coalition logic and cluster input control logic. Each of the plurality of processing cores is assigned to one of a plurality of core clusters and each of the one or more coalition logic is associated with each of the plurality of core clusters for enabling dynamic heterogeneous polymorphic processing by a coalition of two or more of the plurality of processing cores within the associated one of the plurality of core clusters. Each of the one or more coalition logic switches the two or more of the plurality of processing cores into a coalition in response to the one of the plurality of core clusters receiving a coalition request command. The cluster input control logic is associated with each of the plurality of core clusters and receives a processing task comprising a plurality of serial threads, divides each of the plurality of serial threads into a plurality of basic blocks, each of the plurality of basic blocks including a single entry point and a single exit point, and creates a control flow graph for the plurality of basic blocks. The cluster input control logic further adds one of a plurality of sentinel instructions to each of the plurality of basic blocks of the ones of the plurality of serial threads that require more than one processing core, each of the plurality of sentinel instructions including information for processing of the plurality of basic blocks.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The accompanying figures, where like reference numerals refer to identical or functionally similar elements throughout the separate views and which together

with the detailed description below are incorporated in and form part of the specification, serve to illustrate various embodiments and to explain various principles and advantages in accordance with a present invention.

[0014] FIG. 1, comprising FIGs. 1A and 1B, illustrates conventional homogeneous multi-core architectures, wherein FIG. 1A depicts a multiple simple core architecture and FIG. 1B depicts a multiple complex core architecture.

[0015] FIG. 2 illustrates a conventional heterogeneous multi-core architecture.

[0016] FIG. 3 illustrates a dynamic polymorphic multi-core processor architecture in accordance with the present embodiment.

[0017] FIG. 4 pictorially depicts basic block processing by the processor architecture of FIG. 3 in accordance with the present embodiment.

[0018] FIG. 5 depicts a sentinel instruction format in accordance with the present invention.

[0019] FIG. 6 includes graphs depicting the percentages of basic blocks with number of live-in and live-out registers below a threshold percentage in accordance with the present embodiment.

CLAIMS

What is claimed is:

1. A method for dynamic heterogeneous polymorphic processing in a processor comprising a plurality of processing cores, each of the plurality of processing cores assigned to one of a plurality of core clusters and each of the plurality of core clusters dynamically forming a coalition comprising two or more processing cores of the plurality of processing cores, the method comprising the steps of:

receiving a processing task comprising a plurality of serial threads;

dividing each of the plurality of serial threads into a plurality of basic blocks, each of the plurality of basic blocks comprising a single entry point and a single exit point;

creating a control flow for the plurality of basic blocks;

determining whether each of the plurality of serial threads requires more than one processing core;

sending a go-into-coalition-mode-now instruction to ones of the plurality of core clusters for handling ones of the plurality of serial threads that require more than one processing core;

adding one of a plurality of sentinel instructions to each of the plurality of basic blocks of the ones of the plurality of serial threads that require more than one processing core, each of the plurality of sentinel instructions comprising information for processing of the plurality of basic blocks; and

upon execution completion of each of the ones of the plurality of serial threads that requires more than one processing core, sending a go-out-of-coalition-mode-now instruction to corresponding ones of the plurality of core clusters.

2. The method in accordance with Claim 1 wherein the determining step comprises determining whether each of the plurality of serial threads requires more than one processing core in response to complexity of instruction-level parallelism of each of the plurality of serial threads.

3. The method in accordance with Claim 2 wherein the determining step comprises determining that one of the plurality of serial threads requires only one processing core in response to determining that the one of the plurality of serial threads comprises a low instruction level parallelism serial thread.

4. The method in accordance with Claim 2 wherein the determining step comprises determining that one of the plurality of serial threads requires a coalition of two processing cores in response to determining that the one of the plurality of serial threads comprises a medium instruction level parallelism serial thread.

5. The method in accordance with Claim 2 wherein the determining step comprises determining that one of the plurality of serial threads requires a coalition of four processing cores in response to determining that the one of the plurality of serial threads comprises a high instruction level parallelism serial thread.

6. The method in accordance with Claim 1 wherein each of the plurality of sentinel instructions comprises end-with-branch data.

7. The method in accordance with Claim 1 wherein each of the plurality of sentinel instructions comprises length of basic block data.

8. A dynamic heterogeneous polymorphic processor comprising:

a plurality of processing cores, each of the plurality of processing cores assigned to one of a plurality of core clusters;

one or more coalition logic associated with each of the plurality of core clusters for enabling dynamic heterogeneous polymorphic processing by a coalition of two or more of the plurality of processing cores within the associated one of the plurality of core clusters, each of the one or more coalition logic switching the two or more of the plurality of processing cores into a coalition in response to the one of the plurality of core clusters receiving a coalition request command; and

cluster input control logic associated with each of the plurality of core clusters, the cluster input control logic receiving a processing task comprising a plurality of serial threads, dividing each of the plurality of serial threads into a plurality of basic blocks wherein each of the plurality of basic blocks comprises a single entry point and a single exit point, and creating a control flow graph for the plurality of basic blocks,

wherein the cluster input control logic associated with each of the plurality of core clusters further determines whether processing of each of the plurality of serial threads requires one processing core or a coalition of processing cores in response to complexity of instruction-level parallelism of each of the plurality of serial threads,

wherein the cluster input control logic determines that processing of one of the plurality of serial threads requires two processing cores in response to determining that the one of the plurality of serial threads comprises a medium instruction level parallelism serial thread, the cluster input control logic sending a go-into-coalition-mode-now instruction to the associated one of the plurality of core clusters for the associated one of the plurality of core clusters to form a coalition of two of the plurality of processing cores of the associated one of the plurality of core clusters and

passing the one of the plurality of serial threads to coalition logic associated with the coalition.

9. The dynamic heterogeneous polymorphic processor in accordance with Claim 8 wherein the cluster input control logic determines that processing of one of the plurality of serial threads requires one processing core in response to determining that the one of the plurality of serial threads comprises a low instruction level parallelism serial thread, the cluster input control logic passing the one of the plurality of serial threads to one of the plurality of processing cores in the associated one of the plurality of core clusters.

10. The dynamic heterogeneous polymorphic processor in accordance with Claim 8 wherein the cluster input control logic determines that processing of one of the plurality of serial threads requires four processing cores in response to determining that the one of the plurality of serial threads comprises a high instruction level parallelism serial thread, the cluster input control logic sending a go-into-coalition-mode-now instruction to the associated one of the plurality of core clusters for the associated one of the plurality of core clusters to form a coalition of four of the plurality of processing cores of the associated one of the plurality of core clusters and passing the one of the plurality of serial threads to coalition logic associated with the coalition.

11. A dynamic heterogeneous polymorphic processor comprising:
a plurality of processing cores, each of the plurality of processing cores assigned to one of a plurality of core clusters;

one or more coalition logic associated with each of the plurality of core clusters for enabling dynamic heterogeneous polymorphic processing by a coalition of two or more of the plurality of processing cores within the associated one of the plurality of core clusters, each of the one or more coalition logic switching the two or more of the plurality of processing cores into a coalition in response to the one of the plurality of core clusters receiving a coalition request command; and

cluster input control logic associated with each of the plurality of core clusters, the cluster input control logic receiving a processing task comprising a plurality of serial threads, dividing each of the plurality of serial threads into a plurality of basic blocks wherein each of the plurality of basic blocks comprises a single entry point and a single exit point, and creating a control flow graph for the plurality of basic blocks, and

wherein the cluster input control logic associated with each of the plurality of core clusters further determines whether processing of each of the plurality of serial threads requires one processing core or a coalition of processing cores in response to complexity of instruction-level parallelism of each of the plurality of serial threads,

wherein the cluster input control logic determines that processing of one of the plurality of serial threads requires four processing cores in response to determining that the one of the plurality of serial threads comprises a high instruction level parallelism serial thread, the cluster input control logic sending a go-into-coalition-mode-now instruction to the associated one of the plurality of core clusters for the associated one of the plurality of core clusters to form a coalition of four of the plurality of processing cores of the associated one of the plurality of core clusters and passing the one of the plurality of serial threads to coalition logic associated with the coalition.

12. A dynamic heterogeneous polymorphic processor comprising:

a plurality of processing cores, each of the plurality of processing cores assigned to one of a plurality of core clusters;

one or more coalition logic associated with each of the plurality of core clusters for enabling dynamic heterogeneous polymorphic processing by a coalition of two or more of the plurality of processing cores within the associated one of the plurality of core clusters, each of the one or more coalition logic switching the two or more of the plurality of processing cores into a coalition in response to the one of the plurality of core clusters receiving a coalition request command; and

cluster input control logic associated with each of the plurality of core clusters, the cluster input control logic receiving a processing task comprising a plurality of serial threads, dividing each of the plurality of serial threads into a plurality of basic blocks wherein each of the plurality of basic blocks comprises a single entry point and a single exit point, and creating a control flow graph for the plurality of basic blocks,

wherein the cluster input control logic further adds one of a plurality of sentinel instructions to each of the plurality of basic blocks of the ones of the plurality of serial threads that require more than one processing core, each of the plurality of sentinel instructions comprising information for processing of the plurality of basic blocks.

13. The dynamic heterogeneous polymorphic processor in accordance with Claim 12 wherein each of the plurality of sentinel instructions comprises end-with-branch data, length of basic block data, live-in registers and live-out registers.

14. The dynamic heterogeneous polymorphic processor in accordance with Claim 13 wherein each of the plurality of sentinel instructions comprises three or less live-in registers and three or less live-out registers.