What is claimed is:

1. A method comprising:

accessing (210) a computer program, wherein the computer program comprises a conditional statement that comprises a Boolean condition, wherein the Boolean condition takes a first variable A and a second variable B;

transforming (220) the Boolean condition into an arithmetic and/or bitwise expression of the Boolean condition, the transforming comprising:

subtracting (410) B from A,

computing (420) an absolute value of a result of the subtracting B from A,

computing (430) an exponent based on the absolute value of a result of the subtracting B from A,

computing (440) an exponential by exponentiation of a base number by the computed exponent, and

computing (450) the arithmetic and/or bitwise expression of the Boolean condition based on a remainder of a division of the exponential by the base number;

generating (230) an arithmetic and/or bitwise expression of the computer program comprising the arithmetic and/or bitwise expression of the Boolean condition in place of the Boolean condition; ~~and~~

providing (240) the arithmetic and/or bitwise expression of the computer program, wherein the arithmetic and/or bitwise expression of the computer program is configured to be executed by an arithmetic and/or bitwise operation unit of a processor; and

executing (250), by the arithmetic and/or bitwise operation unit, the arithmetic and/or bitwise expression of the computer program.

2. The method of claim 1, wherein either:

~~2. The method of claim 1, wherein the arithmetic and/or bitwise expression of the Boolean condition verifies the following:~~

~~if the Boolean condition is verified, the arithmetic and/or bitwise expression of the Boolean condition is equal to 1; and~~

~~if the Boolean condition is not verified, the arithmetic and/or bitwise expression of the Boolean condition is equal to 0.~~

~~3. The method of claim 1, wherein the Boolean condition takes a first variable A and a second variable B, wherein transforming the Boolean condition into an arithmetic and/or bitwise expression of the Boolean condition comprises:~~

~~subtracting B from A;~~

~~computing an absolute value of a result of the subtracting B from A;~~

~~computing an exponent based on the absolute value of a result of the subtracting B from A;~~

~~computing an exponential by exponentiation of a base number by the computed exponent; and~~

~~computing the arithmetic and/or bitwise expression of the Boolean condition based on a remainder of a division of the exponential by the base number.~~

~~4. The method of claim 3, wherein~~ the Boolean condition is A = B, and wherein transforming the Boolean condition into an arithmetic and/or bitwise expression of the Boolean condition comprises:

computing the exponent as the absolute value of the result of the subtracting B from A~~,~~; and computing the arithmetic and/or bitwise expression of the Boolean condition as the remainder of the division of the exponential by the base number; or ~~.~~

~~5. The method of claim 3, wherein~~ the Boolean condition is A ≥ B, and wherein transforming the Boolean condition into an arithmetic and/or bitwise expression of the Boolean condition comprises:

subtracting the result of the subtracting B from A from the absolute value of the result of the subtracting B from A,

~~;~~

computing the exponent as an absolute value of a result of the subtracting the result of the subtracting B from A from the absolute value of the result of the subtracting B from A, and~~; and~~

~~computing the arithmetic and/or bitwise expression of the Boolean condition as the remainder of the division of the exponential by the base number.~~

computing the arithmetic and/or bitwise expression~~6. The method~~ of the Boolean condition as the remainder of the division of the exponential by the base number; or

claim 3, wherein the Boolean condition is A < B, and wherein transforming the Boolean condition into an arithmetic and/or bitwise expression of the Boolean condition comprises:

subtracting the result of the subtracting B from A from the absolute value of the result of the subtracting B from A,

;

computing the exponent as an absolute value of a result of the subtracting the result of the subtracting B from A from the absolute value of the result of the subtracting B from A,

;

computing the exponential by exponentiation of the base number by the exponent,; and

computing the arithmetic and/or bitwise expression of the Boolean condition as 1 minus the remainder of division of the exponential by the base number; or

the Boolean condition is A < B, and wherein transforming the Boolean condition into an arithmetic and/or bitwise expression of the Boolean condition comprises:

subtracting B from A;

computing the arithmetic and/or bitwise expression of the Boolean condition as a borrow value of a result of the subtracting B from A.

3. 7. The method of claim 13, wherein either:

the Boolean condition is A = B, wherein the first and second variables A and B are floating point type data, and wherein transforming the Boolean condition into an arithmetic and/or bitwise expression of the Boolean condition comprises:

computing the exponent as a result of a ceiling function applied to the absolute value of the result of the subtracting B from A,; and

computing the arithmetic and/or bitwise expression of the Boolean condition as the remainder of the division of the exponential by the base number; or

8. The method of claim 3, wherein the Boolean condition is A ≥ B, wherein the first and second variables A and B are floating point type data, and wherein transforming the Boolean condition into an arithmetic and/or bitwise expression of the Boolean condition comprises:

subtracting the result of the subtracting B from A from the absolute value of the result of the subtracting B from A.

;

computing the exponent as a result of a ceiling function applied to an absolute value of a result of the subtracting the result of the subtracting B from A from the absolute value of the result of the subtracting B from A,; and

computing the arithmetic and/or bitwise expression of the Boolean condition as the remainder of the division of the exponential by the base number; or .

9. The method of claim 3, wherein the Boolean condition is A < B, wherein the first and second variables A and B are floating point type data, and wherein transforming the Boolean condition into an arithmetic and/or bitwise expression of the Boolean condition comprises:

subtracting the result of the subtracting B from A from the absolute value of the result of the subtracting B from A,

;

computing the exponent as a result of a ceiling function applied to an absolute value of a result of the subtracting the result of the subtracting B from A from the absolute value of the result of the subtracting B from A,

;

computing the exponential by exponentiation of the base number by the exponent,; and

computing the arithmetic and/or bitwise expression of the Boolean condition as 1 minus the remainder of division of the exponential by the base number.

4. 10. The method of claim 13, wherein the first and second variables A and B are Boolean type data, wherein the Boolean condition is A = B, and wherein transforming the Boolean condition into an arithmetic and/or bitwise expression of the Boolean condition comprises:

computing a sum of A and B; and

computing the arithmetic and/or bitwise expression of the Boolean condition as 1 minus a remainder of division of the sum by the base number.

5. ~~11.~~ The method of claim 1~~3~~, wherein the first and second variables A and B comprise matrix type data, and the method comprises:

splitting each of the first and second variables A and B into respective first two-element pairs of the first variable A and second two-element pairs of the second variable B; and

forming smaller versions of each of the first and second variables A and B, for comparison, using representative numbers from a mathematical sequence with an increasing difference between members of the mathematical sequence;

wherein each of the representative numbers for the smaller version of the first variable A is computed from respective members of the mathematical sequence having indexes in the mathematical sequence that are equal to data values in the first two-element pairs of the first variable A;

wherein each of the representative numbers for the smaller version of the second variable B is computed from respective members of the mathematical sequence having indexes in the mathematical sequence that are equal to data values in the second two-element pairs of the second variable B; and

wherein, when each of the first and second variables A and B have an odd number of elements, removing a single element from each of the first and second variables A and B before the splitting, and adding the single element from each of the first and second variables A and B to the smaller versions of the first and second variables A and B respectively.

6. ~~12.~~ The method of claim 5~~11~~, wherein the forming comprises:

computing the representative numbers for the smaller version of the first variable A by summing, for each element of the smaller version of the first variable A, two of the respective members of the mathematical sequence that correspond to the data values in the first two-element pairs of the first variable A; and

computing the representative numbers for the smaller version of the second variable B by summing, for each element of the smaller version of the second variable B, two of the respective members of the mathematical sequence that correspond to the data values in the second two-element pairs of the second variable B.

7. ~~13.~~ The method of claim 5~~11~~, comprising sorting each of the first and second variables A and B, wherein the sorting of a matrix-type-data variable comprises:

instantiating a new variable with a same number of elements as the matrix-type-data variable;

dividing the matrix-type-data variable into a number of sub-parts equal to a number of processing threads to be used to perform the sorting;

determining, by each of the respective processing threads, a position index for each element in the sub-part handled by the processing thread by comparing a value of each element of the sub-part with all other values in the matrix-type-data variable and counting how many of the elements of the matrix-type-data variable are less than, or are greater than for an inversed position index, the compared element of the sub-part;

replacing values in the new variable with values from the matrix-type-data variable in accordance with the position indices determined by the processing threads, in either ascending or descending order; and

replacing any remaining unchanged values in the new variable with either a left-adjacent element value or a right-adjacent element value in the new variable, depending on whether the sorting is in ascending or descending order;

wherein the new variable is used as a sorted version of the matrix-type-data variable.

8. ~~14.~~ The method of claim 5~~11~~, comprising:

computing first position variables for the first two-element pairs of the first variable A; and

computing second position variables for the second two-element pairs of the second variable B;

wherein computing each position variable for a pair of elements comprises exponentiation of a number, which is equal to negative 1, by a result of Boolean condition Y > X, wherein X is the first element in the pair of elements, and Y is the second element in the pair of elements; and

wherein the forming comprises

computing the representative numbers for the smaller version of the first variable A by, for each element of the smaller version of the first variable A, multiplying one of the first position variables by a sum of two of the respective members of the mathematical sequence that correspond to the data values in the first two-element pairs of the first variable A, and

computing the representative numbers for the smaller version of the second variable B by, for each element of the smaller version of the second variable B, multiplying one of the second position variables by a sum of two of the respective members of the mathematical sequence that correspond to the data values in the second two-element pairs of the second variable B.

9. ~~15.~~ The method of claim 1~~3~~, wherein the base number is equal to 2, wherein the processor uses a binary system, and wherein the remainder of division of the exponential by the base number is extracted by taking a last bit of the exponential; and ~~.~~

optionally,~~16. The method of claim 3, wherein the processor uses a binary system, and~~ wherein the exponentiation of the base number by the computed exponent is performed by shifting a bit, which is equal to 1, left by a number of places equal to the computed exponent.

10. ~~17. The method of claim 1, wherein the Boolean condition is A < B, and wherein transforming the Boolean condition into an arithmetic and/or bitwise expression of the Boolean condition comprises:~~

~~subtracting B from A;~~

~~computing the arithmetic and/or bitwise expression of the Boolean condition as a borrow value of a result of the subtracting B from A.~~

~~18.~~ The method of claim 1, wherein providing the arithmetic and/or bitwise expression of the computer program comprises:

fragmenting the arithmetic and/or bitwise expression of the computer program into multiple fragments of arithmetic and/or bitwise expression; and

wherein the fragments of arithmetic and/or bitwise expression are configured to be executed in parallel.

11. The method of claim 1, wherein the arithmetic and/or bitwise expression of the Boolean condition verifies the following:

if the Boolean condition is verified, the arithmetic and/or bitwise expression of the Boolean condition is equal to 1, and

if the Boolean condition is not verified, the arithmetic and/or bitwise expression of the Boolean condition is equal to 0; and

~~19. The method of claim 2,~~ wherein the Boolean condition comprises a consequent task that is formulated as an arithmetic statement, and wherein generating the arithmetic and/or bitwise expression of the computer program comprises:

multiplying the consequent task with the arithmetic and/or bitwise expression of the Boolean condition; and ~~.~~

optionally~~20. The method of claim 19~~, wherein the Boolean condition comprises an alternative task that is formulated as an arithmetic statement, and wherein generating the arithmetic and/or bitwise expression of the computer program comprises ~~:~~

multiplying the alternative task with a 1 minus the arithmetic and/or bitwise expression of the Boolean condition.

12. The method of claim 1, wherein the arithmetic and/or bitwise expression of the Boolean condition verifies the following:

if the Boolean condition is verified, the arithmetic and/or bitwise expression of the Boolean condition is equal to 1, and

if the Boolean condition is not verified, the arithmetic and/or bitwise expression of the Boolean condition is equal to 0; and

~~21. The method of claim 2,~~ wherein the Boolean condition comprises a consequent task that is formulated as a non-arithmetic statement, and wherein generating the arithmetic and/or bitwise expression of the computer program comprises:

including in the arithmetic and/or bitwise expression of the computer program an unconditional jump to a label associated with the arithmetic and/or bitwise expression of the Boolean condition; ~~.~~

optionally~~22. The method of claim 21~~, wherein the label is equal to a start of a statement of the consequent task when a value of the arithmetic and/or bitwise expression of the Boolean condition is equal to 1; and ~~.~~

optionally~~23. The method of claim 22~~, wherein the Boolean condition comprises an alternative task that is formulated as a non-arithmetic statement, wherein the label is equal to a start of a statement of the alternative task when the value of the arithmetic and/or bitwise expression of the Boolean condition is equal to 0.

13. The method of claim 1~~24. The method of claim 1, wherein the method further comprises:~~

~~executing, by the arithmetic and/or bitwise operation unit, the arithmetic and/or bitwise expression of the computer program.~~

~~25. The method of claim 24~~, wherein at least the transforming is performed by a first computer, and the executing is performed by a second computer.

14. ~~26.~~ The method of claim 1~~24~~, wherein the transforming and the executing are performed by a same computer.

15. ~~27.~~ A system (100) comprising:

one or more instruction processing units (110) comprising at least one control unit (120),~~;~~ at least one arithmetic and/or bitwise operation unit (130),~~;~~ and at least one register unit (140);~~;~~ and

one or more memory units (150) coupled with the one or more instruction processing units;

wherein the system is configured to perform substantially all conditional statements, which each comprise a Boolean condition, in a computer program by using the at least one arithmetic and/or bitwise operation unit to execute arithmetic and/or bitwise expressions of the Boolean conditions of the substantially all conditional statements; and ~~.~~

~~28. The system of claim 27,~~ wherein the at least one control unit or a separate data processing apparatus of the system is configured to perform method operations of any of claims 1-14~~26, and wherein the at least one arithmetic and/or bitwise operation unit of the processor comprising the one or more instruction processing units is configured to execute the arithmetic and/or bitwise expression of the computer program~~.

16. ~~29.~~ The system of claim 15~~27~~, wherein the at least one arithmetic and/or bitwise operation unit comprises one or more of an arithmetic block (131),~~,~~ a bitwise shift block (132),~~,~~ and a bitwise logical block (133),~~,~~ and wherein the at least one arithmetic and/or bitwise operation unit does not include any digital comparator.

~~30. A method comprising:~~

~~accessing a computer program comprising a conditional statement comprising a Boolean condition that compares a first variable A and a second variable B, wherein each of the first and second variables A and B comprises matrix type data;~~

~~splitting each of the first and second variables A and B into respective first two-element pairs of the first variable A and second two-element pairs of the second variable B; and~~

~~forming smaller versions of each of the first and second variables A and B, for comparison, using representative numbers from a mathematical sequence with an increasing difference between members of the mathematical sequence, wherein each of the representative numbers for the smaller version of the first variable A is computed from respective members of the mathematical sequence having indexes in the mathematical sequence that are equal to data values in the first two-element pairs of the first variable A, each of the representative numbers for the smaller version of the second variable B is computed from respective members of the mathematical sequence having indexes in the mathematical sequence that are equal to data values in the second two-element pairs of the second variable B, and wherein, when each of the first and second variables A and B have an odd number of elements,~~

~~removing a single element from each of the first and second variables A and B before the splitting, and~~

~~adding the single element from each of the first and second variables A and B to the smaller versions of the first and second variables A and B respectively;~~

~~comparing the smaller versions of the first and second variables A and B to evaluate the Boolean condition of the conditional statement when running the computer program.~~

~~31. The method of claim 30, wherein the forming comprises:~~

~~computing the representative numbers for the smaller version of the first variable A by summing, for each element of the smaller version of the first variable A, two of the respective members of the mathematical sequence that correspond to the data values in the first two-element pairs of the first variable A; and~~

~~computing the representative numbers for the smaller version of the second variable B by summing, for each element of the smaller version of the second variable B, two of the respective members of the mathematical sequence that correspond to the data values in the second two-element pairs of the second variable B.~~

~~32. The method of claim 30, comprising:~~

~~computing first position variables for the first two-element pairs of the first variable A; and~~

~~computing second position variables for the second two-element pairs of the second variable B;~~

~~wherein computing each position variable for a pair of elements comprises exponentiation of a number, which is equal to negative 1, by a result of Boolean condition Y > X, wherein X is the first element in the pair of elements, and Y is the second element in the pair of elements; and~~

~~wherein the forming comprises~~

~~computing the representative numbers for the smaller version of the first variable A by, for each element of the smaller version of the first variable A, multiplying one of the first position variables by a sum of two of the respective members of the mathematical sequence that correspond to the data values in the first two-element pairs of the first variable A, and~~

~~computing the representative numbers for the smaller version of the second variable B by, for each element of the smaller version of the second variable B, multiplying one of the second position variables by a sum of two of the respective members of the mathematical sequence that correspond to the data values in the second two-element pairs of the second variable B.~~

~~33. A system comprising:~~

~~one or more instruction processing units comprising at least one control unit, at least one arithmetic and/or bitwise operation unit, and at least one register unit; and~~

~~one or more memory units coupled with the one or more instruction processing units;~~

~~wherein the one or more memory units are configured to cause the one or more instruction processing units to perform method operations of any of claims 30-32.~~

~~34. A method of sorting a matrix-type data object, the method comprising:~~

~~instantiating a new object with a same number of elements as the matrix-type data object;~~

~~dividing the matrix-type data object into a number of sub-parts equal to a number of processing threads to be used to perform the sorting;~~

~~determining, by each of the respective processing threads, a position index for each element in the sub-part handled by the processing thread by comparing a value of each element of the sub-part with all other values in the matrix-type data object and counting how many of the elements of the matrix-type data object are less than, or are greater than for an inversed position index, the compared element of the sub-part, wherein the determining comprises performing the method of any of claims 1-12 and 14-26;~~

~~replacing values in the new object with values from the matrix-type data object in accordance with the position indices determined by the processing threads, in either ascending or descending order; and~~

~~replacing any remaining unchanged values in the new object with either a left-adjacent element value or a right-adjacent element value in the new object, depending on whether the sorting is in ascending or descending order, wherein the new object is used as a sorted version of the matrix-type data object.~~

~~35. A system comprising:~~

~~one or more instruction processing units comprising at least one control unit, at least one arithmetic and/or bitwise operation unit, and at least one register unit; and~~

~~one or more memory units coupled with the one or more instruction processing units;~~

~~wherein the one or more memory units are configured to cause the one or more instruction processing units to perform method operations of claim 34.~~